LEVERAGING MID-LEVEL REPRESENTATIONS FOR COMPLEX ACTIVITY RECOGNITION

A Dissertation
Presented to
The Academic Faculty

By

Unaiza Ahsan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology

May 2019

LEVERAGING MID-LEVEL REPRESENTATIONS FOR COMPLEX ACTIVITY RECOGNITION

Approved by:

Dr. Irfan Essa, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. James Hays
School of Interactive Computing
Georgia Institute of Technology

Dr. Devi Parikh
School of Interactive Computing
Georgia Institute of Technology

Dr. Munmun De Choudhury
School of Interactive Computing
Georgia Institute of Technology

Dr. Zsolt Kira
School of Interactive Computing
Georgia Institute of Technology

Dr. Chen Sun
Senior Research Scientist
Google

Date Approved: November 27, 2018

Dedicated to Chotpo

ACKNOWLEDGEMENTS

First of all, I would like to thank God Almighty for providing me the opportunity and strength to complete this chapter of my life.

My sincere gratitude goes out to my advisor, Dr. Irfan Essa for choosing me to be his Ph.D. student and for providing me with a lot of support throughout my studies at Georgia Tech. His timely suggestion enabled me to apply for a fellowship that ended up being my primary source of funding for five years. All his efforts whenever I needed help are greatly appreciated. Many thanks to Schlumberger Faculty for the Future Fellowship program for funding my studies here at Georgia Tech.

I would like to thank my thesis committee members who helped me immensely in various stages of my Ph.D.. They were supportive and encouraging and never hesitated to provide helpful advice and feedback on my research progress. I would especially like to thank Dr. James Hays in enabling me to use his node on SkyNet server for a very important part of this thesis. Special thanks to my external advisor Dr. Chen Sun for always giving helpful insights into the problems and providing critical feedback all the time.

A huge shout out to my lab colleagues (Aneeq, Patsorn, Huda, Vince, Rapha, Erik, Jon, Shray, Varun, Apoorva, Niranjan, Daniel, Luke, Daniel, Steve, Amirreza and Samarth) who provided not only a lot of support in terms of research, proof-reading and advice, but were also a source of great fun and laughter in the lab. I recall I bugged many of them with questions and emails and they always replied promptly and enabled me to complete this work with ease. I would especially like to mention Varun Agarwal for the role he played in providing crucial feedback and tech support especially when I was trying to get access to our server. Many thanks to the senior students of our lab (especially Vinay and Yachna) who provided major guidance in my early days of the Ph.D. and did not stop helping me out even after they graduated. Special thank you to Kyla Hanson for organizing and scheduling meetings and taking care of logistics for all of us.

I would like to thank my teachers and mentors from school, college and grad school for their hard work and support. I would especially like to mention Dr. Saneeha Ahmed, Dr. Sohail Sattar and Dr. Humera Noor for sparking my interest in AI and Computer Vision and helping me during my Master's thesis.

A sincere thank you to my entire family, for patiently bearing with the long-term nature of Ph.D. and blessing me with their prayers and support. My parents (Dr. Ahsan and Dr. Nasreen) are responsible for raising me, educating me and supporting my interest in higher education. Chotpo (my youngest Aunt) was the driving force behind my motivation for doing a Ph.D. Her insights about research and its requirements (as she herself is a Ph.D. supervisor in my home country, Pakistan) were super useful for me throughout my studies. Her support never wavered for me, despite her several scary health issues in the past six years. My eldest Aunt (Apa) always kept in touch with me (despite her busy schedules) and provided me amazing pictures of her sweet granddaughter, Romaisa. I am indebted to my sister, Hafsa, for always being "an email/text" away - literally - no matter how busy she was. Her children, Ibrahim and Zainab, provided a much-needed relief from stress too many times to count. Special thanks goes to my nephew Ibrahim for patiently waiting for his "Aala" to complete all her "school work" and not forgetting me, despite the fact that I left for my studies when he was just three years old.

I would also like to mention that this work would never (ever) have been possible without the selfless support of my parents-in-law (Shama Aunty and Zafar Uncle) who always provided a listening ear, supportive advice and helpful suggestions to maintain a healthy work-life balance.

Lastly, I would like to say a huge thank you to my husband, Munzir, who was always there for me, every step of this journey and helped me a lot with everything, ranging from offering technical help with algorithm implementation to doing *all* household chores when I was working towards a deadline.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Dynamic scene understanding requires learning representations of the components of the scene including objects, environments, actions and events. Complex activity recognition from images and videos requires annotating large datasets with action labels which is a tedious and expensive task. Thus, there is a need to design a mid-level or intermediate feature representation which does not require millions of labels, yet is able to generalize to semantic-level recognition of activities in visual data. This thesis makes three contributions in this regard.

First, we propose an event concept-based intermediate representation which learns concepts via the Web and uses this representation to identify events even with a single labeled example. To demonstrate the strength of the proposed approaches, we contribute two diverse social event datasets to the community. We then present a use case of event concepts as a mid-level representation that generalizes to sentiment recognition in diverse social event images.

Second, we propose to train Generative Adversarial Networks (GANs) with video frames (which does not require labels), use the trained discriminator from GANs as an intermediate representation and finetune it on a smaller labeled video activity dataset to recognize actions in videos. This unsupervised pre-training step avoids any manual feature engineering, video frame encoding or searching for the best video frame sampling technique.

Our third contribution is a self-supervised learning approach on videos that exploits both spatial and temporal coherency to learn feature representations on video data without any supervision. We demonstrate the transfer learning capability of this model on smaller labeled datasets. We present comprehensive experimental analysis on the self-supervised model to provide insights into the unsupervised pretraining paradigm and how it can help with activity recognition on target datasets which the model has never seen during training.

CHAPTER 1
INTRODUCTION

The widespread adoption of smart-phones coupled with easy to use photo sharing services has resulted in massive user-shared multimedia in the form of photographs, tweets, text, audio and/or video on social networking websites. The success of Convolutional Neural Networks (CNNs) in visual recognition is primarily due to large labeled visual data. However, large labeled datasets like ImageNet [1] are hard to come by because of the effort required in labeling images on a large scale. The problem worsens if we consider videos as they comprise a set of frames which have to be individually labeled in order to get meaningful video-level annotations and therefore, the success of deep learning approaches for image categorization tasks has been hard to replicate on videos. Video annotation is expensive and for fine-grained tasks such as action localization, hard to even agree upon [2]. One may argue that social media contains text as well as images and text can thus be used to automatically label images and videos. However, text-based analysis on social platforms has its own challenges because of variations in style, punctuation, grammar, vocabulary and syntax [3]. Recent attempts to generate labels for visual data automatically by the Computer Vision community include large scale datasets like Sports1M [4] and YouTube-8M [5] but the challenge remains to map noisy labels to useful annotations in order to use CNNs effectively for activity recognition. We address the problem of learning from weak/noisy labels in this thesis by leveraging the use of mid-level representations that incorporates information that is essentially free or can be obtained with minimum effort but generalizes to high level recognition of activities and events from images and videos even with limited labeled examples. The major questions we address in this thesis are:

1. Can we design an intermediate concept-level representation for images using just information from the Web and transfer it to one-shot event recognition from images?

2. Can we generalize the concept-space for abstract recognition application such as image sentiment recognition?

3. Moving on to videos, can we exploit the inherent structure of the video data to design a self-supervised task that uses no labels but generalizes to activity recognition in vidoes?

1.1 Importance of Mid-level Representation for Visual Recognition

The neurons in the human visual cortex are arranged in layers [6] and recent Computational Neuroscience studies have shown that Bag-of-Words (BoW)-based representations are formed in the middle layers [7]. These intermediate representations then lead to high level object and scene recognition in the visual cortex. The design and usage of mid-level representations for visual recognition has a long history in Computer Vision. These feature representations are highly useful for recognizing those categories for which many labeled examples are not available.

There are several ways of designing a mid-level representation for visual recognition, namely, part-based models [8], attribute-based [9] and hierarchical (CNN)-based models [10]. One important mid-level feature representation for visual recognition is the use of attributes or concepts. Attributes have been used to describe objects (both fixed [11, 9, 12, 13] and relative [14]), faces [15], scenes [16] and actions [17, 18]. These attribute detectors are then run on new images for high level recognition [19, 20]. Researchers have explored creating a set (or bank) of detectors pretrained on objects such as Object Banks [21], an ontology of abstract concepts such as Classemes [22] or scene attributes [16, 23].

Leveraging an attribute-based intermediate representation not only benefits us by requiring fewer labeled examples, it also provides *semantic meaning* to scenes involving complex activities and events. Motivated by this research, our first contribution is to design an intermediate concept space to recognize social events from images without requiring large-scale labeled examples. To demonstrate the strength of the proposed approaches, we

contribute two diverse social event datasets to the community. We then present a use case of event concepts as a mid-level representation that generalizes to sentiment recognition in social event images.

1.2 Unsupervised Learning for Activity Recognition in Videos

Since there are even more limitations to producing large labeled video datasets, the community has approached this problem in several ways. For recognizing activities from few examples, many approaches use concepts as intermediate representations for event recognition [24, 25]. Another proposed solution to this problem is self-supervised learning where auxiliary tasks are designed to exploit the inherent structure of unlabeled datasets and a network is trained to solve those tasks. Generative models such as the largely popular Generative Adversarial Networks (GANs) [26] approximate high dimensional probability distributions like those of natural images using an adversarial process without requiring expensive labeling. Our next contribution is to leverage a GAN trained on video frames as an intermediate representation which can then be finetuned on a labeled video dataset for activity recognition.

There are several studies that empirically validate that the earliest visual cues captured by infants' brains are surface motions of objects [27]. These then go on and develop into perception involving local appearance and texture of objects. [27]. Studies have also pointed out that objects' motion and their temporal transformation are important for the human visual system to learn the structure of objects [28, 29]. Motivated by these studies, there is recent work on unsupervised video representation learning via tracking objects through videos and training a Siamese network to learn a similarity metric on these object patches [30]. However, the prerequisite of this approach is to track millions of objects through videos and extract the relevant patches. Keeping this in mind, our next contribution is to propose to learn such a structure of objects and their transformations over time by designing a self-supervised task which solves jigsaw puzzles comprising multiple video

frame patches, without needing to explicitly track objects over time. Our self-supervised method, trained on a large scale video activity dataset also does not require optical flow based patch mining and we show empirically that a large unlabeled video dataset with a simple permutation sampling approach are enough to learn an effective unsupervised representation that generalizes to activity recognition, object recognition in still images as well as unsupervised dicovery (clustering) of activities in videos.

1.3 Thesis Contributions

In this thesis, our contributions are:

- **Event concept-based intermediate representation**: We propose an event concept-based intermediate representation which learns concepts via the Web and uses this representation to identify complex events in images even with a single training example.

- **Sentiment recognition using event concepts**: We demonstrate the discriminative power of event concept features by learning sentiments of event images using concepts outperforming the state-of-the-art in sentiment recognition in images.

- **Contributed datasets**: To demonstrate the strength of the proposed approaches, we contribute two diverse social event datasets to the community as well as a dataset of event images annotated with sentiment labels.

- **GAN-based intermediate representation**: We leverage Generative Adversarial Networks as a mid-level representation to learn actions from videos and demonstrate in preliminary results that our approach performs comparably to the state-of-the-art in semi-supervised video action recognition.

- **Self-supervised spatiotemporal intermediate representation**: We propose a novel self-supervised task which divides multiple video frames into patches, creates jigsaw

puzzles out of these patches and the network is trained to solve this task. We show via extensive experimental evaluation the feasibility and effectiveness of our approach on video action recognition. We propose a permutation strategy that constrains the sampled permutations and outperforms random permutations while being memory efficient. Our work exploits both spatial and temporal context in one joint framework without requiring explicit object tracking in videos or optical flow based patch mining from video frames.

- **Domain transfer capability**: We demonstrate the domain transfer capability of our proposed video jigsaw network, given that our best self-supervised model is trained on Kinetics [31] video frames and we demonstrate competitive results on UCF101 [32] and HMDB51[33] datasets.

1.4 Thesis Organization

The rest of this thesis is organized as follows:

In Chapter 2 we propose a technique to identify social events from images with limited training examples by designing an intermediate concept space and utilizing that for event recognition in one-shot learning setting. In Chapter 3, we present an application of the aforementioned event concept space by demonstrating its effectiveness in reocognizing event sentiments from images. In Chapter 4, we present our approach and results when using a trained discriminator from GANs for video action recognition. In Chapter 5, we present our self-supervised approach for activity recognition in videos. In Chapter 6, we conclude the thesis by presenting results on unsupervised activity recognition using the video jigsaw model weights without finetuning on the target dataset i.e. clustering videos into activities and thus explore the limits of self-supervised learning.

CHAPTER 2
COMPLEX EVENT RECOGNITION FROM IMAGES WITH FEW TRAINING EXAMPLES

In this chapter, we propose to leverage concept-level representations for complex event recognition in photographs given limited training examples. We introduce a novel framework to discover event concept attributes from the web and use that to extract semantic features from images and classify them into social event categories with few training examples. Discovered concepts include a variety of objects, scenes, actions and event sub-types, leading to a discriminative and compact representation for event images. Web images are obtained for each discovered event concept and we use (pretrained) CNN features to train concept classifiers. Extensive experiments on challenging event datasets demonstrate that our proposed method outperforms several baselines using deep CNN features directly in classifying images into events with limited training examples. We also demonstrate that our method achieves the best overall accuracy on a dataset with unseen event categories using a single training example.

2.1 Introduction

The recent success of deep Convolutional Neural Networks (CNNs) in object and scene recognition has resulted due to large labeled training databases such as ImageNet [1] and Places [34]. Current approaches which use pretrained CNNs and fine-tune on datasets also require significant number of labeled examples. Since creating huge labeled datasets from the constantly evolving space of events is not realistic, we propose to learn an event concept-based representation and leverage that to identify rare events. Discovering web-driven concepts using Wikipedia and Flickr tags, we aim to categorize social events from static photographs when few labeled examples are available. Images of social events inher-

Figure 2.1: Event concepts as an intermediate feature representation for recognizing social events in photographs.

ently consist of a combination of objects (e.g. 'banner'), scenes (e.g. 'ground'), actions (e.g. 'shouting slogans'), event subtypes (e.g. 'speech') and attributes (e.g. 'protest peacefully'). Object appearance significantly changes when combined with different objects, actions and attributes in cluttered backgrounds. Hence recognizing events from static images requires us to explicitly learn concept classifiers, each concept a combination of objects, scenes, actions and attributes. We call these *event concepts* (see Figure 2.1).

Event recognition approaches that use concepts or attributes have previously been applied to video-based events [25, 35, 36, 37] where temporal dynamics play an important role in recognizing what is happening in the video. This makes event recognition from a single photograph an interesting and challenging problem domain. Several attribute-based recognition methods require datasets annotated with all the concepts [11, 15] which is a tedious process. To bypass the need for manual concept labeling and inspired by the recent 'webly supervised' learning approaches [38, 39, 40] that use web content to discover visual concepts, we propose an event concept learning framework using Wikipedia to gen-

erate event categories and Flickr tags as our initial pool of concepts. From noisy Flickr tags, we generate segments or phrases using a tweet segmentation algorithm proposed by Li *et al.* [41] which is a method designed specifically to extract event-centric phrases from noisy twitter streams. Finally, we project each event category on to a word embedding pretrained on the Google News Dataset using the popular *word2vec* [42] approach, extract nearest neighbors and add them to the pool of segmented phrases. We extract images related to each concept from MS Bing image search engine and compute deep CNN [43] features extracted from a pretrained network on all the images and train concept classifiers. The concept scores predicted on a given test image form the final features for event images.

Our **primary contributions** are:

1. A novel framework which involves using web data to discover event related concepts and employing efficient concept pruning strategies that result in clean, relevant and diverse event concepts.
2. A concept-based representation that not only improves single-shot event classification performance but can also be generalized to those categories which were not used during concept discovery.
3. A large scale Social Event Image Dataset (SocEID) comprising 37,000 general event images belonging to 8 event categories as well as a challenging Rare Events Dataset (RED) comprising 7,000 images belonging to 21 specific real world events.

2.2 Related Work

A recent line of work proposes learning visual concepts from the Web with minimal human supervision ('webly supervised approaches'). NEIL [38] uses image search engine results in a semi-supervised setting to learn and train visual concept detectors. LEVAN [39] uses Google NGram corpus to extract all possible words related to a given concept, extracts images from image search engine and learn visual concepts related to any given keyword. The authors of [44] use a multiple instance learning approach to learn concepts from image

search results. Some approaches learn concepts from images and their labels [45], from image descriptions [46] or by using a deep network [40] using principles from curriculum learning. Our proposed work is inspired by web supervision but for a different domain. The key difference between our approach and other webly supervised concept learning approaches is that our methods are designed to obtain *event specific* concepts. We explain further in Section 2.3.

The earliest work addressing event classification from static images [47] classifies sports events (rowing, rock climbing etc.) using object and scene information. Some related approaches [48, 49] require scene geometry or temporal alignment between event images to identify individual events. Recently [50] propose to train two deep networks; one on images and the second one on spatial maps of detected people/objects at different scales for event recognition. Our work aims to learn relevant concepts from the web and uses pretrained CNNs for feature extraction thus saving training time. More importantly, training a deep network for identifying events in images requires a large labeled dataset. We attempt to eliminate that requirement by discovering and using general event concepts from the web.

Motivated by the need to learn with few labeled examples, vision researchers have addressed one-shot learning to learn object classifiers [51, 52, 53, 54, 55] and more recently used deep networks [56, 57] and part-based models [58]. Ma *et al.* [37] use labels in external videos as concepts and jointly model concept classification and event detection. Chen *et al.* [59] use Flickr tags to discover concepts for an event and its associated text description. Cui *et al.* [60] propose Concept Bank which consists of events mined from WikiHow and concepts from Flickr tags. Ye *et al.* [35] extend the previous work and propose to arrange events and their concepts in a hierarchy learned from WikiHow articles and YouTube descriptions. Shao *et al.* [61] generate video event attributes using crowd-based annotation and use motion channels and appearance to train a deep model. Yang *et al.* [62] learn video concepts from YouTube descriptions and Flickr tags. They generate event concepts using

a tweet segmentation algorithm along with other metrics and train multiple classifiers for a single concept. The major difference between their work and ours is that we begin from event labels instead of descriptions, target image-based event recognition when few labeled examples are available and integrate word2vec based concepts into our concept pool.

2.3 Approach

We begin by asking the question, "What if all events had limited labeled examples?" In reality, there are several events for which labeled image datasets are available e.g. birthdays and weddings. We develop methods and test our approach on datasets containing these popular events by taking only a single labeled example from each category as training data. The rest of the dataset is used for testing. This approach enables us to determine whether our proposed methods work on popular events before taking on the harder task of identifying rare events (using our RED Dataset) from images.

Our proposed method is an adaptation of the webly supervised learning approaches to learn visual concepts relevant to specific event categories. Extracting all visual concepts related to a keyword such as 'birthday' from the web using the approach of [39] results in many concepts that have either little to do with a birthday event or is not generalizable to real world images. Those concepts include birthday settings advertised by event planners online and objects associated with birthday with a clean background and a canonical viewpoint [63]. Hence we argue for event-specific concepts for complex event recognition from images with few labeled examples.

Our approach is divided into three main parts: Event Concept Discovery, Training Concept Classifiers and Prediction of Concept Scores for Event Classification.

2.3.1 Event Concept Discovery

We use Wikipedia to mine a list of events from its category 'Social Events'. This list contains general events (such as birthday) and specific events (such as royal wedding). We

Table 2.1: Sample events mined from Wikipedia

air shows	**auto show**	**beauty pageants**
all star games	**ballet show**	**beer festivals**
american football match	**ballroom dance**	**birdwatching**
annual protests	**balls**	**black friday**
art exhibition	**barbecue**	**boating**
arts festivals	**baseball**	**bowling**
astronomy events	**basketball match**	**boxing**

do *not* include specific events in our initial events list as the aim is to build a concept bank that is applicable for all events. If we build a concept bank for royal wedding, concepts such as 'Kate Middleton,' 'Buckingham Palace' etc. would be too specific to apply to generic event images. Thus we end up with 150 generic social events (see Table 2.1). This list of events was mined from Wikipedia but filtered (to remove specific events) manually.

Each event category in our list is used to query Flickr and we obtain the first 200 image results. We collect the tags (a set of words describing the image) of those images in the form of captions and this forms the noisy caption pool from which we aim to generate meaningful concepts that can describe general social events. We empirically observe that by increasing the number of images from which we mine Flickr tags, the noise in the data increases hence we limit ourselves to 200 images per event category.

Tag Segmentation: For the set of events $E = \{e_1, e_2, ...e_n\}$ where, for our work $n = 150$, we have a set of tags $T = \{t_1, t_2, ...t_N\}$, $N = 200n$ in our case. Our goal is to generate consecutive and non-overlapping segments $S = \{s_1, s_2, ...s_m\}$. These segments can be a single word or phrases. We obtain a set of segments $S_i = \{s_1, s_2, s_3, ...s_{m_i}\} \subset S$ for each tag $t_i \in T$, $i \in \{1, ..., N\}$, by applying a tweet segmentation method [64] which can be modeled as an optimization,

$$\underset{s_1,s_2,s_3,...s_{m_i}}{\operatorname{argmax}} \; Stk(t_i) = \sum_{j=1}^{m_i} Stk(s_j), \tag{2.1}$$

Figure 2.2: Generated segments from Flickr tags for event label 'protest.'

where $Stk(\cdot)$ is a function that computes the *stickiness* of a segment. Stickiness measures the probability that a segment of text is a 'named entity' (name of a person, place or object) in a large text corpus or a knowledge base like Wikipedia. The final score for each segment s_j is given by

$$\text{score}(s_j) = Stk(s_j) \cdot V_{\text{flickr}}(s_j). \tag{2.2}$$

In Equation 2.2 Stk(s_j) refers to the stickiness score of the jth segment and V_{flickr} refers to *visual representativeness* which is a measure of how visually coherent a word or phrase is. Words like 'economy,' and 'public,' if queried to an image search engine, result in ambiguous images, whereas a phrase like 'birthday cake' generally returns very similar images. Hence it has a high visual representativeness score as compared to words like 'economy' or 'activity.' We obtain visual representativeness scores for each segment via a public dataset made available by Sun and Bhowmick [65] where they provide representativeness scores for the most popular tags on Flickr. After computing the final scores, we inspect the highest scoring segments to remove ambiguous or slang words. Figure 2.2 shows some sample tags and the returned high scoring segments. Further details on tag segmentation can be found in [64].

Generating segments from tags is not enough to cover all aspects of a complex event. To expand the taxonomy and find event-specific concepts we additionally project each

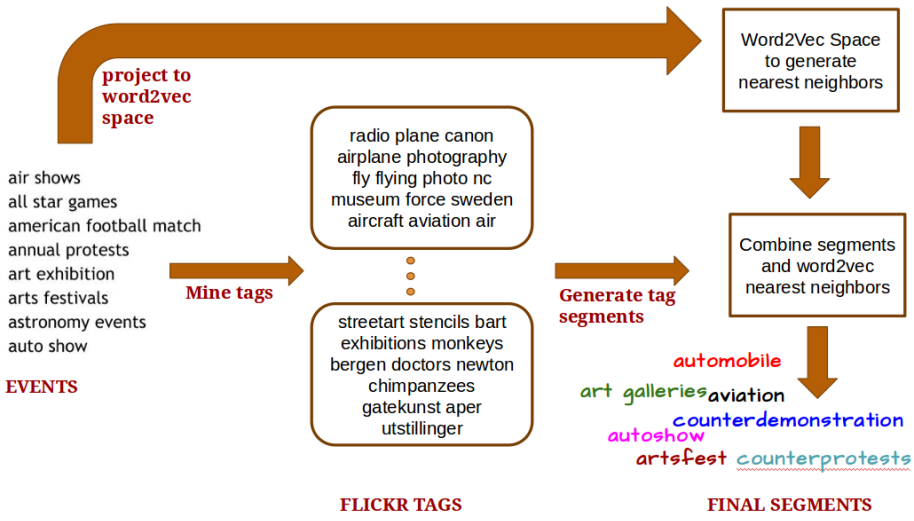

Figure 2.3: Event concept discovery pipeline for generic social events.

event label to word2vec space [42]. We use an embedding trained on the Google News Dataset which consists of about 100 billion words. The model is available for public use (https://code.google.com/p/word2vec) and contains 300-dimensional vectors for 3 million words and phrases. The advantage of using Google News pretrained vectors is that we obtain semantically similar concepts for each event label. Table 2.2 shows the nearest neighbors of the event label 'dance' in word2vec space.

We extract top 20 nearest neighbors for each event label and add them to the tag segments generated via the segmentation scheme described above. This pool of event concepts is then filtered to remove duplicate concepts, slang words and foreign words. We finally end up with 856 event concepts. Our concepts not only include objects, scenes and actions but also include sub-events and their types. Our event concept discovery pipeline is shown in Figure 2.3.

Table 2.2: Nearest neighbors of the event 'dance' in word2vec space

dance	**breakdancing, salsa dancing, argentinean tango, wows crowd, freerunning, reggae hiphop, disco fever, flash mobs, street dance, dancers, breakbeat, hop, dance craze, dancefest, bollywood bhangra, asian pop, dance workout, hip hop dance troupe, breakdancers**

2.3.2 Training Concept Classifiers

Now we describe our approach for selecting training images for concept classifiers. Given a set of concepts $C = \{c_1, c_2, \ldots c_m\}$, we input each concept as an image search query to Microsoft Bing and retrieve the top 100 images returned for each concept. Clipart, duplicate images and images containing only text are removed from the search results. Several concepts in C are correlated with each other, resulting in similar images for different concepts. Training them independently without taking into account their correlation will lead to false negatives that will impact the concept classifier training negatively. The reason why we do not exclude correlated concepts in our concept pool C is that they capture different (and thus important) aspects of a social event. For example, 'birthday party,' 'birthday boy' and 'birthday celebrations' are three different concepts within the same event category (birthday) and they may have similar images. Thus when selecting training images for training the classifier for concept c_i, it is naive to sample negative training images from all concepts c_j where $j \neq i$. Figure 2.4 shows an example of correlated concepts and their associated images.

Hence, we first cluster all the concepts using their word2vec-based vector representations using minibatch k-means clustering [66]. We set $k = 150$. Thus for ith concept c_i, belonging to a y-sized cluster, we obtain a list of concepts $C^i_{pos} = \{c_{i,pos_1}, c_{i,pos_2}, \ldots c_{i,pos_y}\}$ that are highly correlated with the given concept c_i. We construct the concept classifier training set for concept c_i as follows:

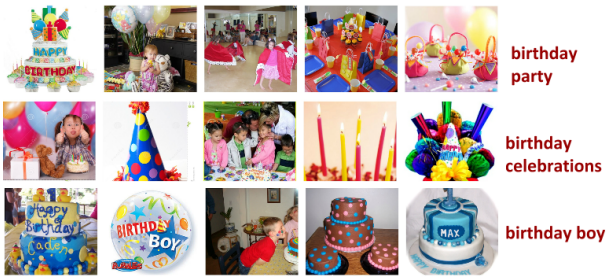

Figure 2.4: Examples of correlated event concepts

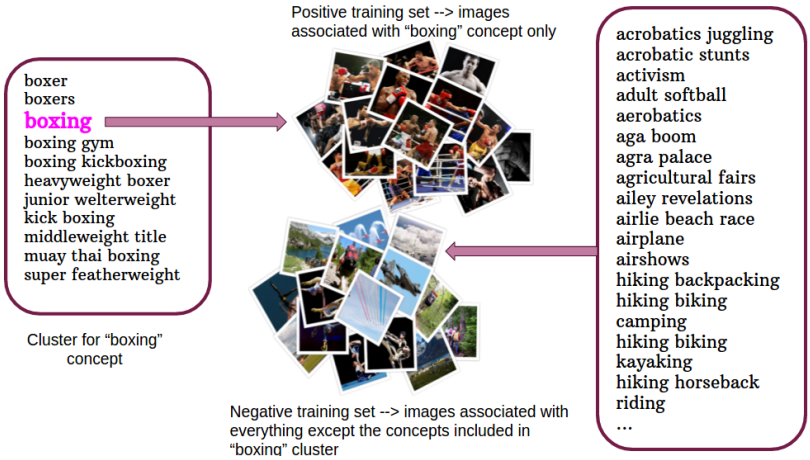

Figure 2.5: Selecting training images for 'boxing' classifier

- Let $\xi^+ = \{I_i\}_{i=1}^{u}$ be the set of positive training images for classification where u is the number of images retrieved for concept c_i.

- Let $\xi^- = \{I_j\}_{j=1}^{v}$ be the set of negative training images for classification where v is the number of images retrieved for the set of concepts $C_{neg}^i = \overline{C_{pos}^i} = C - C_{pos}^i$.

In other words, we make sure that the set ξ^- does not include images retrieved for any concept in the set C_{pos}^i because those images are highly similar to the images in ξ^+ as they belong to the same cluster as c_i (See Figure 2.5).

For each concept, we extract the CNN 'fc7' layer activations as features from all its images, select the training and test examples as described above and input them to logistic regression classifiers. We select the classifier parameters through 5-fold cross validation and for all of our concept classifiers, the cross validation accuracy is above 90%.

2.3.3 Predicting Concept Scores for Classification

After training all concept classifiers, we compute the classifier scores on images belonging to our evaluation datasets. For each image I, its feature vector is a concatenation of all concept classifier scores predicted on the image. Thus $f_I = \{x_i\}_{i=1}^{m}$ where m is the total number of concepts and x_i is the score predicted for ith concept classifier. Finally, we use

these features to classify event images into social events using a linear SVM with default parameters fixed for all experiments. We outline our experimental setup in detail in the next section.

2.4 Experiments and Evaluations

We evaluate our approach on four labeled event datasets with one-shot learning, that is, we use a single positive and negative training example from each class. We also report classification results for all-shot learning (using a 70-30 split) for comparison.

2.4.1 Datasets

We evaluate our concept-based event recognition algorithm on the following four datasets:

1. Social Event Image Dataset (SocEID): This is the dataset we created in-house. We collected images of the following social events: birthdays, graduations, weddings, marathons/races, protests, parades, soccer matches and concerts. We queried Instagram and Flickr with a tag related to the event itself ('wedding day,' 'Graduation 2014' etc.) and downloaded public images in chronological order determined by post date. Our dataset includes some relevant images from the NUS-WIDE dataset [67] and the Social Event Classification subtask from MediaEval 2013 [68]. We passed all the images to 3 trusted coders (non-Turkers) and asked them to filter any image that did not depict a particular social event. We established the final scores on the images via majority vote and discarded all the rest. Finally, we ended up with nearly 37,000 images. Figure 2.6 shows sample images from the SocEID dataset.

2. Web Image Dataset for Event Recognition (WIDER): This dataset is introduced by [50] and consists of 50,574 images annotated with 61 classes. The classes are as follows: Parade, Handshaking, Demonstration, Riot, Dancing, Car Accident, Funeral, Cheering, Election Campain, Press Conference, People Marching, Meeting, Group, Interview, Traffic, Stock Market, Award Ceremony, Ceremony, Concerts, Couple, Family Group, Festival,

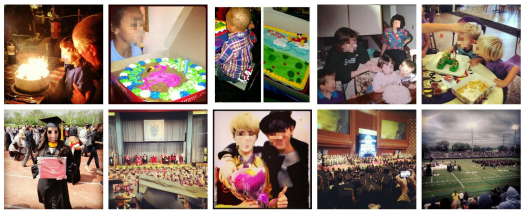

Figure 2.6: Sample images of the SocEID Dataset for two events: birthday (top) and graduation (bottom).

Figure 2.7: Sample images of the Rare Events Dataset.

Picnic, Shoppers, Soldier Firing, Soldier Patrol, Soldier Drilling, Spa, Sports Fan, Students Schoolkids, Surgeons, Waiter Waitress, Worker Laborer, Running, Baseball, Basketball, Football, Soccer, Tennis, Ice Skating, Gymnastics, Swimming, Car Racing, Row Boat, Aerobics, Balloonist, Jockey, Matador Bullfighter, Parachutist Paratrooper, Greeting, Celebration Or Party, Dresses, Photographers, Raid, Rescue, Sports Coach Trainer, Voter, Angler, Hockey, People Driving Car and Street Battle. Our full results on the WIDER dataset are shown in Table 2.4.

3. UIUC Sports Event Dataset: This dataset [47] consists of 1579 images belonging to 8 sports events categories which are: badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snowboarding. Our method outperforms the baselines in all the categories for one-shot learning.

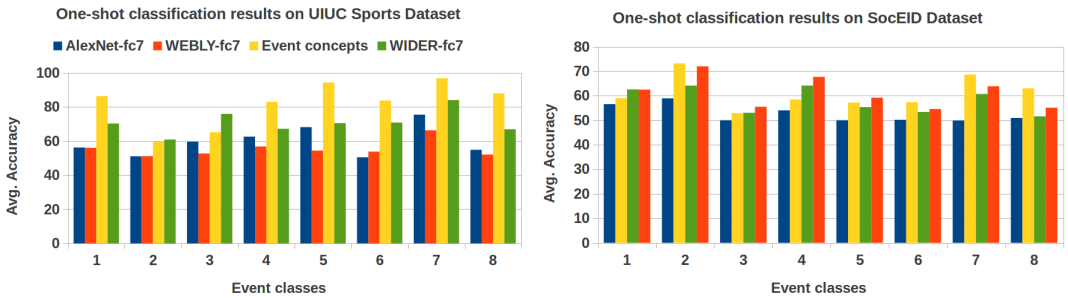

Figure 2.8: One-shot learning results on UIUC Sports Dataset and SocEID.

4. Rare Events Dataset (RED): This is another in-house dataset we collected by querying MS Bing image search engine with a set of 26 'rare' event categories (see Figure 2.7). We call them rare not on the basis of how frequently they occur in the world but on how seldom they are found in large labeled event image datasets. The full list of event categories are as follows Russian airstrikes, Boston bombing, Nepal earthquake, Arab Spring, 9-11 attacks, Russian airlines crash Sinai, Paris attacks, 2012 summer olympics, drone attacks, Hurricane Katrina, Mali attacks, Boston Red Sox win 2004, Columbia Space Shuttle disaster, election campaign Trump, Humanity washed ashore, US invasion Afghanistan, Yemen civil war, Barack Obama wins elections 2008, Hurricane Sandy, Israel Palestine conflict and Justin Trudeau elected. The whole dataset comprises nearly 7,000 images and we do not remove any image from any event category manually. Note that these are all *specific* events and our main motivation behind collecting this dataset is twofold: i) Since few labeled examples are available for these events, it is a suitable test case for our claim that our learned concepts are a powerful intermediate representation to recognize events with few examples, ii) We want to test whether our discovered event concepts generalize to recognizing specific event images or not.

Table 2.3: Top 10 predicted concepts for sports events 'rowing' and 'polo'

rowing	rowing championships, rowing regatta, recreational boating, canoe trip, junior rowing, canoe polo, standup pandling, swimming canoeing, recreational fishing, cable wakeboard
polo	horseback riding, horse show, mountain biking, bronc riding, bareback bronc, horseback ride, ranch rodeo, stampede rodeo, canoeing horseback, riding mountain

Table 2.4: Result of one-shot learning on WIDER Dataset

Features	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AlexNet-fc7	57.51	55.76	50.94	**53.26**	50.45	**65.23**	52.70	51.56	53.66	55.50	53.01	52.38	54.53	**57.69**	58.86
WEBLY-fc7	57.97	55.54	53.05	53.07	57.43	64.59	**54.32**	**54.26**	**55.46**	**60.19**	56.03	**60.07**	**55.13**	56.31	**60.70**
Event concepts	**60.46**	**56.62**	**63.64**	52.16	**59.63**	64.96	51.71	51.59	54.80	56.87	**57.82**	59.29	49.00	57.05	59.47
Features	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
AlexNet-fc7	51.17	55.55	52.89	54.16	56.43	54.45	50.86	55.75	**55.25**	56.81	59.19	54	61.84	55.06	50.47
WEBLY-fc7	**59.14**	59.08	54.67	59.1	55.01	**56.35**	**58.8**	**62.58**	54.7	61.52	59.51	**60.87**	**67.59**	53.73	52.23
Event concepts	57.55	**59.2**	**60.22**	**59.63**	**57.36**	54.1	56.29	61.4	54.33	**64.12**	**63.5**	57.05	63.35	**56.15**	**54.92**
Features	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
AlexNet-fc7	62.87	56.86	51.51	53.95	54.75	49.58	55.93	54.97	57.33	57.93	58.84	59.51	**78.74**	67.22	54.73
WEBLY-fc7	**64.14**	59.52	53.37	60.54	59.12	52.67	**63.18**	60.69	**69.8**	**64.06**	62.67	57.23	64.4	67.04	54.49
Event concepts	57.51	**59.94**	**53.44**	**64.00**	**61.01**	**54.37**	52.22	**64.61**	61.94	60.07	**71.11**	**68.12**	70.74	**72.34**	**58.98**
Features	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
AlexNet-fc7	52.08	49.39	54.66	55.78	51.84	51.96	60.88	48.75	53.01	51.19	54.56	51.8	59.95	54.59	55.13
WEBLY-fc7	**55.76**	52.82	**55.56**	**58.7**	**52.33**	56.32	63.42	53.41	**56.1**	52.71	**55.6**	52.97	**66.86**	57.23	**61.9**
Event concepts	55.69	**55.36**	52.94	53.47	50.1	**57.02**	**64.16**	**54.24**	54.78	**54.69**	51.31	**56.54**	64.17	**57.64**	56.81
Features	61	overall													
AlexNet-fc7	52.4	55.40													
WEBLY-fc7	**53.47**	58.15													
Event concepts	52.01	**58.29**													

2.4.2 Experimental Setup

We begin our experiments by training event concept classifiers. We have a total of 86,000 images associated with the concepts, retrieved from Microsoft Bing using the publicly available Bing crawler by Dengxin Dai.[1] We use the Caffe [69] deep learning framework to extract CNN layer 7 activations ('fc7') as features for all the images using HybridCNN which is a publicly available CNN model pretrained on 978 object categories from ImageNet database [1] and 205 scene categories from Places dataset [34] using the AlexNet deep architechture [10]. For each concept, we select the positive and negative training features as described in Section 2.3.2 and train L2-regularized logistic regression classifiers using the publicly available LIBLINEAR library [70]. Every image in our event datasets is input to each of the trained concept classifiers and a probabilistic concept score is computed on it. The fusing of concept scores form the final feature vector of that image.

[1] `http://www.vision.ee.ethz.ch/~daid`

Training with a Single Positive Image

We conduct our one-shot learning experiment on the event datasets as follows: For event category E with P positive training features and N negative training features we randomly sample a positive training feature f_p^E and a negative training feature f_n^E. We concatenate the two and feed this into a binary linear SVM as training features. For testing, we simply take the rest of the positive features ($P - f_p^E$) and sample an equal number of negative features from the rest of the event categories in the dataset. Hence for all our experiments, the random baseline is 50%. We run all experiments five times and average the per-class classification accuracies.

Training with a 70%-30% split

In this experiment we take all of the labeled data into account and for each class, randomly select 70% of images for training and test on the remaining images. This experiment shows the maximum accuracy our method can achieve given all of the training data available. It provides a nice comparison against the case where only a single labeled image is available for each class. We compare our results with several powerful baselines:

- AlexNet [10] pretrained on ImageNet [1] and Places [34] databases, from which we extract 4096-dimensional layer fc7's activations and use them as features. We refer to this baseline as AlexNet-fc7 in the results.

- Chen *et al.* [40] which is a recently proposed webly supervised CNN trained on about 2.1 million images downloaded from Google Images using popular vision datasets' labels as search queries. The authors use 2,240 objects, 89 attributes, and 874 scene labels from ImageNet [1], SUN database [71] and NEIL knowledge base [38] and use principles from curriculum learning to train the network with easy examples first and then hard examples from Flickr. They show state of the art performance compared to AlexNet for objection detection and scene recognition. We use their 'GoogleA'

Table 2.5: Result of one-shot learning on RED Dataset

Event classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Avg. Acc
AlexNet-fc7	54.0	53.4	60.0	**56.8**	53.3	54.0	53.0	50.3	59.9	**56.0**	52.5	57.5	65.3	54.0	**53.6**	53.2	55.5	55.5	**64.0**	50.7	62.6	56.0
WEBLY-fc7	52.2	58.1	60.9	56.2	53.9	**56.3**	50.4	55.5	67.8	54.9	**61.0**	**61.3**	59.6	55.6	47.8	**61.0**	53.2	52.1	56.0	52.6	**69.0**	57.0
WIDER-fc7	50.4	55.5	57.2	52.2	**55.8**	55.1	51.1	52.8	55.5	52.6	52.6	58.9	63.3	53.7	47.3	49.8	53.4	**55.6**	61.3	52.8	59.3	54.6
Event concepts	**57.5**	**58.9**	**67.7**	55.5	54.8	53.1	**53.9**	**58.6**	**75.1**	54.2	60.8	55.4	**73.4**	**56.2**	52.0	58.0	**56.2**	52.4	58.7	**53.3**	64.8	**58.6**

network which is trained on 2.1 million Google images. We refer to this baseline as WEBLY-fc7.

- AlexNet [10] finetuned on WIDER database [50]. This baseline model is provided by the authors. We want to see how fc7 features extracted from this model perform with limited training data on our evaluation datasets (except WIDER) and whether our concept-level features are comparable to it. We refer to this baseline as WIDER-fc7.

2.5 Results and Discussion

Our one-shot learning result on UIUC Sports Events dataset (Figure 2.8 left) shows that the event concept features significantly outperform all the baselines in 6 out of 8 events. From 1-8, the UIUC Sports event categories are: badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snowboarding. The main reason why this occurs is that our initial event list and hence our discovered event concepts include several sports events and their subtypes. For example, for all the images labeled with the event "rowing" and "polo" in the UIUC Sports Dataset, we count the top 10 most frequently predicted concepts. Table 2.3 qualitatively shows that our method extracts relevant concepts consistently accross the set of UIUC Sports events images.

Our one-shot learning experiment on the SocEID dataset (Figure 2.8 right) results in event concepts outperforming the baselines in 4 out of 8 categories. From 1-8, the categories are: birthday, concert, graduation, marathons, parade, protest, soccer and wedding. This is very likely due to the nature of images found in our dataset. Our dataset contains very clean images from the Web with significant visual cues of popular events such as birthdays. Thus the webly supervised network of [40] and WIDER [50] finetuned on event

images is able to discriminate between the different event classes based on cues such as graduation caps and bridal gowns in the graduation and wedding pictures respectively even if limited training examples are present.

Our experiments on the WIDER dataset [50] yield interesting insights. There are 61 classes in the dataset and we follow the order given by the authors. We test one-shot learning using our event concept features and compare them against the baselines, WEBLY-fc7 and AlexNet-fc7. Our event concept features outperform AlexNet-fc7 and WEBLY-fc7 in 31 classes when training with a single image. Table 2.4 shows our results. We note that the WIDER dataset contains not only events but also individual actions such as 'cheering'. Our proposed approach uses concepts such as 'cheering' to identify events which typically involve cheering such as dances, games or graduations (as the cheering concept will result in high probabilistic prediction on such events). However, the model is not trained to identify cheering alone. This can be verified by noting our performance scores in classes such as parade (1), demonstration (3), dancing (5) etc.. We score well on other categories such as basketball, soccer, running, aerobics etc. Categories where our scores are comparable but not above the baselines are: sports coach trainer, greeting, surgeons, spa and stock market, to name a few. These categories are recognized more effectively by deep CNNs pretrained to recognize objects and scenes.

Finally, we evaluate our method on the RED dataset (see Table 2.5) which is the most challenging because the images are highly diverse and consist of specific real world events. In one-shot learning on RED, for 10 out of 21 classes, our proposed method outperfoms the baselines. Thus our method is able to generalize to unseen event categories as our initial event list does not contain any of the rare event categories in the RED dataset. From 1 to 21, the categories are: (we have marked in bold those categories on which our method outperforms all the baselines for one-shot learning): **Russian airstrikes**, **Boston bombing**, **Nepal earthquake**, Arab Spring, 9-11 attacks, Russian airlines crash Sinai, **Paris attacks**, **2012 summer olympics**, **drone attacks**, Hurricane Katrina, Mali attacks, Boston Red Sox win

Table 2.6: Overall accuracy for one-shot learning on the evaluation datasets

	Overall Average Accuracy (%)			
Features	UIUC Sports	SocEID	WIDER	RED
AlexNet-fc7	59.79	52.57	55.40	55.94
WEBLY-fc7	55.39	60.89	58.14	56.92
WIDER-fc7	70.81	58.11	N/A	54.58
Event concepts	**82.09**	**62.98**	**58.29**	**58.59**

Table 2.7: Overall accuracy for all-shot learning on the evaluation datasets

	Overall Average Accuracy (%)			
Features	UIUC Sports	SocEID	WIDER	RED
AlexNet-fc7	96.47	**86.42**	77.94	77.86
WEBLY-fc7	95.16	83.66	77.85	**79.39**
WIDER-fc7	93.85	80.42	N/A	76.64
Event concepts	**96.68**	85.39	**78.59**	77.57

2004, **Columbia Space Shuttle disaster**, **election campaign Trump**, Humanity washed ashore, US invasion Afghanistan, **Yemen civil war**, Barack Obama wins elections 2008, Hurricane Sandy, **Israel Palestine conflict** and Justin Trudeau elected.

The overall classification accuracies for one-shot learning are shown in Table 2.6. For all the datasets, event image classification using a single training example with our proposed event concepts as features outperform the baselines which shows the strength of our approach to recognize complex real world events when limited labeled examples are available.

We also evaluate our method against the baselines using all the available training data (70%-30% split). Table 2.7 shows the overall classification accuracies on our evaluation datasets. Even when using all training examples, our event concept features are comparable to the state of the art in recognizing events from images. For two datasets, (UIUC Sports and WIDER) our method actually outperforms the state of the art.

We also show some qualitative results on top five predicted concepts on random test images. See Figures 2.9, 2.10 and 2.11.

Figure 2.9: Top 5 predicted concepts for a random wedding event image from the SocEID Dataset.

Figure 2.10: Top 5 predicted concepts for a random running event image from the WIDER Dataset.

2.6 Summary

In this chapter we propose to discover event-specific concepts from the web to recognize complex events from images with few labeled examples. Our proposed framework discovers relevant concepts by combining segmented Flickr tags and word2vec nearest neighbors of event categories resulting in a compact intermediate representation which identifies real

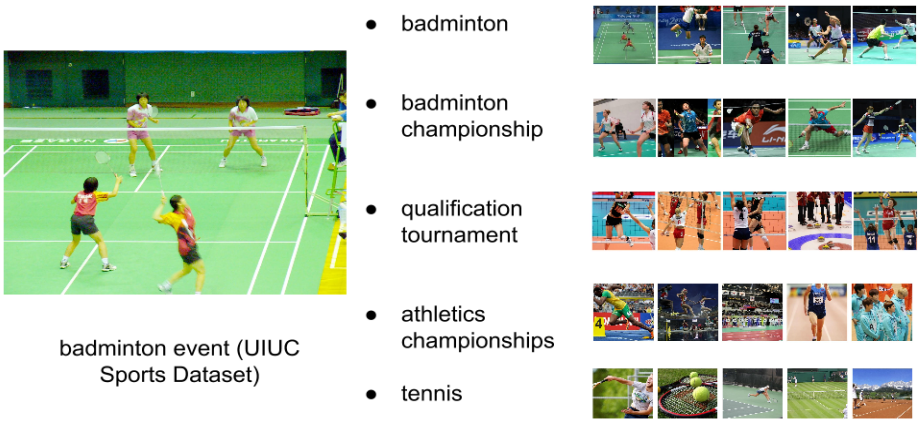

Figure 2.11: Top 5 predicted concepts for a random badminton event image from the UIUC Sports Dataset.

world events with only a single training example. We show the strength of our proposed method by evaluating on challenging datasets against powerful baselines which directly use CNN features pretrained on objects, scenes, attributes and events. It is interesting to note that in the problem domain of event recognition from visual content where only a few training examples are available, web-driven concept discovery and web images for training can result in highly discriminative intermediate representations which outperform directly using deep CNNs trained on millions of images and even deep CNNs finetuned on a large event dataset.

CHAPTER 3

APPLICATION: EVENT SENTIMENT RECOGNITION VIA ATTRIBUTES

In this chapter, we propose to capture sentiment information of such social event images leveraging their visual content. Our method extracts an intermediate visual representation of social event images based on the visual attributes that occur in the images going beyond sentiment-specific attributes. We map the top predicted attributes to sentiments and extract the dominant emotion associated with a picture of a social event. Unlike recent approaches, our method generalizes to a variety of social events and even to unseen events, which are not available at training time. We demonstrate the effectiveness of our approach on a challenging social event image dataset and our method outperforms state-of-the-art approaches for classifying complex event images into sentiments.

3.1 Introduction

Social media platforms such as Instagram, Flickr, Twitter and Facebook have emerged as rich sources of media, a large portion of which are images. Instagram reports that on average, more than 80 million photos are uploaded daily to its servers.[1] This includes images of personal major life events such as weddings, graduations, funerals, as well as of collective news events such as protests, presidential campaigns and social movements. While some images are usually accompanied with associated text in the form of tags, captions, tweets or posts, a large part of visual media does not contain meaningful captions describing the image content or labels describing visual affect.

Inference of psychological attributes such as sentiment from text is well-studied [72], however the extraction of sentiment via the visual content of images remains underexplored. Recent approaches that infer visual sentiment are limited to images containing an object, person or scene [73]. We address the problem of inferring the dominant affect of

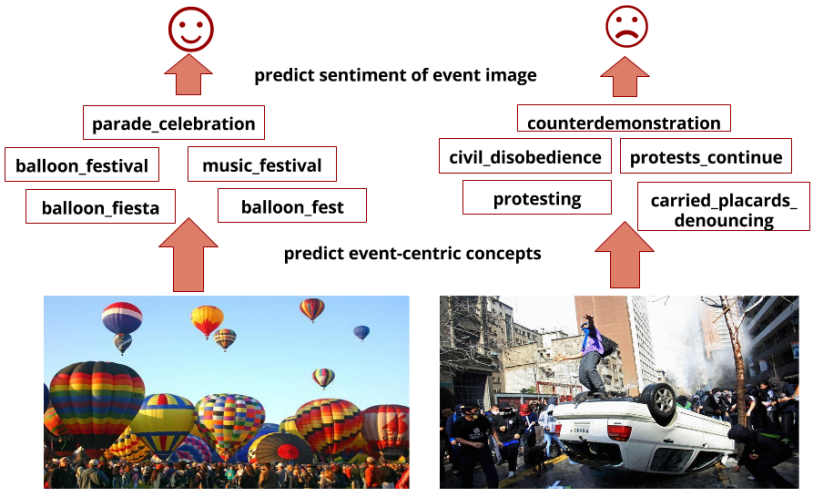

Figure 3.1: Our major contribution is to map event concepts to sentiments for social event images.

a photograph containing complex and often crowded scenes that characterize many social and news events. Our goal is to use only visual features of the given photograph and not rely on any metadata (See Figure 3.1).

Our motivation to use only visual data for sentiment prediction springs from three observations. (1) Automatically predicting sentiments on event images can help determine what users feel about the event and in what context they choose to share it online. This can help personalize social feeds of individuals, as well as improve recommendation algorithms. (2) News events are often shared in the form of collated articles with images. Accurately ascertaining the sentiment of the specific event images using text will lead to inherent biases that may be introduced by the text or caption of the image. (3) Text associated with an event image may not convey sufficient, accurate or reliable sentiment related information. For instance, some tags or captions may just describe the objects, actions or scenes occurring in the image without reflecting on the actual emotional state conveyed through the image.

Event images usually consist of objects (e.g. wedding gown, cake), scenes (e.g. church), people (e.g. bride), subevents (e.g. ring exchange), actions (e.g. dancing) and the like. We

[1]https://instagram.com/press, accessed April 2016

refer to these as *event concepts*. They are similar to the mid-level representations in sentiment prediction pipelines referred to as adjective noun pairs (ANPs) (e.g. cute baby, beautiful landscape) but there are no explicit adjectives or sentiments in our event concepts. In this chapter we develop a sentiment detection framework that infers complex event image sentiment by exploiting visual concepts on event images. Our method discovers concepts for events and extracts intermediate representation of event images using probabilistic predictions from concept models [74].

Concretely, the contributions of our work are:

- We propose a method to predict the sentiment of complex event images using visual content and event concept detector scores without requiring any text analysis on test images.
- Our method outperforms state-of-the-art sentiment prediction approaches without extracting sentiment specific information from the images.
- We conduct comprehensive experiments on a challenging social event image dataset annotated with sentiment labels (*positive, negative, neutral*) from crowdworkers, and propose to share this dataset with the research community.
- To assess generalizability and validity, we employ our event sentiment detector on a large dataset of web images tagged with events *not considered* in model training, and characterize the nature of sentiments expressed in them.

3.2 Related Work

The increased use of social media by people in the last decade resulted in research opportunities to determine what people feel and emote about entities and events. Twitter emerged as a powerful platform to share opinions on daily events. Prior work includes developing frameworks to analyze sentiments on predidential debates [75, 76], SemEval Twitter sentiment classification task [77, 78] and brands [79]. De Choudhury *et al.* mapped moods

into affective states [80] and also predicted depression from social media posts [81]. In attempts to make sense of large-scale community behavior, Kramer *et al.* utilized the text of posts made on Facebook to determine social contagion effects of emotion and affect [82]; whereas Golder and Macy [83] found that positive and negative affect expressed on Twitter can replicate known diurnal and seasonal behavioral patterns across cultures. All these approaches use text as a major source of sentiment discovery. We address the problem of identifying emotions conveyed by complex event images, without reliance on associated text.

Recent work on emotion prediction from images or videos leveraged low level visual features [84, 85, 86], user intention [87], attributes [73, 88], art theory-based descriptors [89] and face detection [90]. Our work is similar to the SentiBank [73] approach which extracts sentiment concepts-based representation of images and then predicts their sentiment using the concept representation as features but our method differs in one crucial way. We do not extract sentiment-related concepts on images such as 'cute baby' but event-related concepts such as 'birthday boy'. Hence our representation differs as it is *event specific* and not sentiment specific. Wang *et al.* [91] used web images and associated text to jointly learn image sentiment using a nonnegative matrix factorization approach. Our work differs from theirs in terms of image type. They predicted sentiment on images where objects and faces are clearly visible (hence dedicated object/scene/face detectors can be used). We focus on event sentiment detection from crowded event images where faces and objects may not be clearly visible.

Other similar work includes methods using deep networks for sentiment prediction but differ in that they either use sentiment specific features [92, 93], do not use intermediate concepts [94] or use probabilistic sampling to select training instances with discriminative features [95]. All of these methods do not address sentiment prediction of images containing complex and crowded scenes. A more recent line of work has started addressing emotion recognition in group images/videos [96, 97, 98, 99, 100, 101] however our prob-

lem domain is different as we do not require human beings or their faces to be visible in the image in order to predict the sentiment of the image.

3.3 Approach

In this section we present our sentiment classification framework starting from the proposed event concepts. Our method comprises three main steps: (1) Generating event concepts, (2) Computing event concept scores, and (3) Predicting sentiment labels from concept scores.

We first discover event concepts by mining an initial list of event categories from Wikipedia. Those categories are then used as search queries to mine Flickr tags. Thereafter, using a tweet segmentation algorithm [41] on these noisy tags, we generate generate relevant social event concepts. Finally, we combine these discovered concepts with nearest neighbors obtained by projecting event categories onto a semantic vector space (word2vec) [42]. For each discovered event concept, we crawl images shared on the web, compute convolutional neural network (CNN) features on them and train concept models. Once the models are trained, we predict concept scores on test images to compute our proposed features and finally use a linear Support Vector Machine (SVM) to predict the sentiment of the test images.

3.3.1 Generating Event Concepts

Using a concept-based intermediate representation as image features is an established technique for capturing high level semantic information from images [84, 85, 86]. Our main motivation behind generating event specific concepts is to formulate a discriminative representation for crowded event images using web-based results and social media tags. Off-the-shelf deep CNN features are useful for object and scene recognition from images but directly using these features for classifying sentiment of crowded event images is not sufficient due to the inherent ambiguity and complexity associated with visual manifestation of affect (as will also be illustrated in the results section).

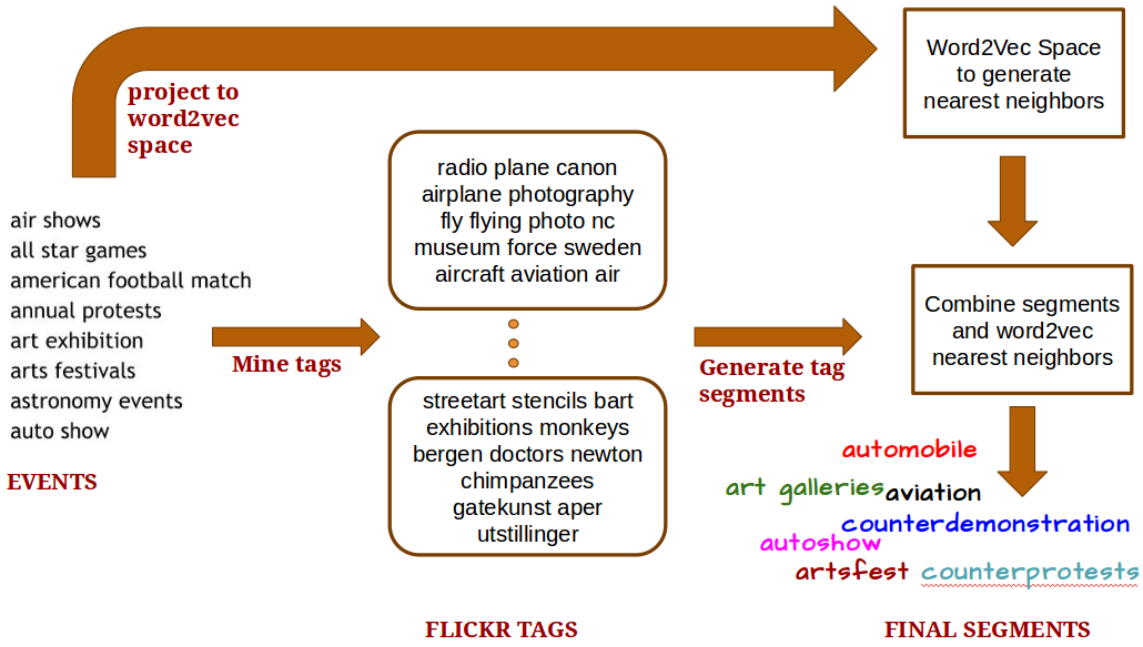

Figure 3.2: Generating event concepts for social events [74]

We generate relevant social event concepts using the following steps:

1. We use Wikipedia to mine a list of 150 social event categories from its category 'Social Events'. This list is generic in order to cover all possible types and categories of events. Some sample event categories are: basketball match, art festivals, beauty pageants, black friday etc..

2. We use the event categories as exact queries to Fickr and retrieve top 200 tags for public images.

3. We preprocess the tags and employ them to a tweet segmentation algorithm proposed by [41] to generate coherent segments (phrases). This algorithm uses a dynamic programming approach to select only those combination of words that have high probability of occurence in large text corpuses and words that are named entities. We also make sure the extracted segments are visually representative [65]. We inspect the highest scoring segments after computing the final scores and remove ambiguous or slang words.

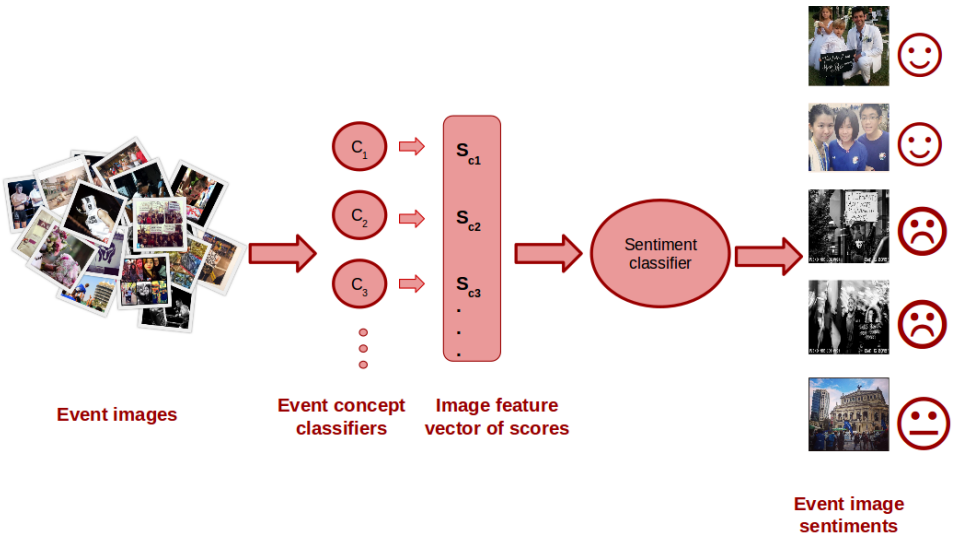

Figure 3.3: Sentiment classification pipeline.

4. Finally, we project each event category (mined from Wikipedia) on to a word embedding using the popular word2vec [42] approach. The word embedding is pre-trained on the Google News Dataset—a large corpus of text from Google News articles comprising around 100 billion words. We extract 20 nearest neighbors to each event category and add them to the pool of segmented phrases. We use the word vectors pretrained on Google News Dataset because as it is a collection of words from news articles, the word vectors refer to those words and phrases which involve news events and are hence relevant to our work. After pruning irrelevant concepts, we finally end up with 856 social event concepts. Figure 3.2 shows the event concept discovery pipeline. For further details, please see [74].

3.3.2 Computing Event Concept Scores

Each generated event concept is used as a search query on the Microsoft Bing search engine to extract the top 100 public images. MS Bing is a convenient platform for scraping highly discriminative images for a wide variety of search queries. The images are used to train linear classifiers to predict concept scores on our test images. The image features used are

the activations of the last layer (fc7) in a Convolutional Neural Network (CNN) pretrained on ImageNet [1] and Places Databases [34] and the CNN architecture used is AlexNet [10][1]. We compute fc7 features on each image and use event concept classifiers to predict the concept probabilistic scores. For each image I, the feature vector f_I is a concatenation of all concept classifier scores predicted on the image. Thus $f_I = \{x_i\}_{i=1}^{m}$ where m is the total number of concepts and x_i is the score predicted for ith concept classifier. In our proposed method, $m = 856$.

3.3.3 Predicting Sentiment Labels

Given that event concepts generated from similar images are likely to be semantically similar, our hypothesis is that these concepts would capture the sentiment conveyed in the image. For example, a birthday event image may contain top predicted concepts such as 'celebrations', 'party' *etc*. These are all positive concepts and thus, the overall image is predicted to be a positive image, as opposed to neutral or negative. Event concepts can thus predict the emotion conveyed by the image without any explicit sentiment-related feature computation. Figure 3.3 shows the complete event image sentiment classification pipeline.

3.4 Experiments

In this section we describe our event image dataset, the user study conducted to generate sentiment labels for the dataset and our experimental setup to predict event image sentiments on the test set.

3.4.1 Dataset

We retrieve public images from Microsoft Bing using 24 event categories as search queries. Our event categories include **accidents**, **airplane crash**, **baby shower**, **birthday**, **carnivals**, **concerts**, **refugee crises**, **funerals**, **wedding**, **protests**, **wildfires**, **marathons etc.**

[1]Hybrid-CNN model is publicly available at https://github.com/BVLC/caffe/wiki/Model-Zoo

These events are diverse, capture both planned and unplanned events and include personal as well as community-based events. We obtain around 10,500 images. We pass these images to the crowdsourcing platform Amazon Mechanical Turk and request crowdworkers to rate the sentiment of each image. We ask them to mark images with **one** of the following five options: (1) Positive, (2) Negative, (3) Neutral, (4) Not an event image or (5) Image does not load. Each image is labeled by three crowdworkers. We accept responses only from those workers who are located in the US and who have an approval rating of more than 95%.

We build our event sentiment database based on the following rules:

- We only keep images if at least 2 out of 3 crowdworkers agree on its sentiment label, whether positive, negative or neutral.

- We discard all images on which fewer than 2 crowdworkers agree on the sentiment label of the event image. We also discard those images crowdworkers mark as 'Not an event image' and 'Image does not load'.

We discard images on which crowdworkers disagree because of the subjective nature of the task. The final number of images retained is 8,748. Hence we find that crowdworkers agree on the sentiment labels of 83.3% of the initial images.

The distribution of sentiments in our final dataset is shown in Figure 3.4. As the pie chart shows, the positive and neutral images are more than six times as many as the negative images. This is because social media platforms are generally perceived as places that promote the sharing and dissemination of positive thoughts and behaviors. Further, the recent Facebook emotional contagion study [82], pointed to the fact that people engage more with positive posts, while negative posts decrease user engagement. Hence, even for events that are negative in general (such as earthquakes, societal upheavals and crises), images related to rehabilitation efforts, political liberty or community solidarity may be perceived as positive.

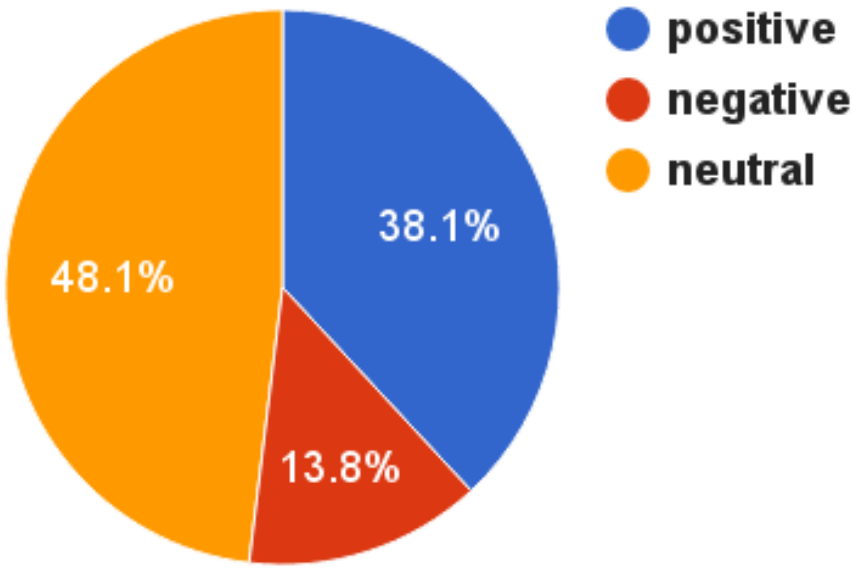

Figure 3.4: Distribution of sentiments in our crowd-annotated social event image dataset.

Figure 3.5 shows a few examples of positive, negative and neutral images as annotated and agreed upon by crowdworkers. The top row shows positive images and it can be seen that many different events can convey positive emotions. Similarly, negative images show clear cases of violence and attacks. The bottom row shows neutral events and this is what the bulk of the images are annotated as; as no clear positive or negative emotion is conveyed by these images.

3.4.2 Experimental Setup

We set up our experiments with the annotated event image dataset. For training, we randomly sample 70% of the images from each sentiment class as positive training data and an equal number of training images from the rest of the sentiment classes as negative training data. We test on the remaining (30%) of images per class. Our test set also consists of an equal number of negative test data sampled from the other sentiment classes than the one being tested. Hence our sentiment prediction baseline accuracy is always 50%. We use

Figure 3.5: Event images with sentiments agreed upon by majority vote: The top row shows positive event images, middle row shows negative images and bottom row shows neutral images.

this one-vs-all strategy, repeat this procedure 5 times and average the sentiment prediction accuracies per class to obtain the final accuracy.

We compute our event concept scores on the images by using the Caffe [69] deep learning framework. This tool extracts CNN layer 7 activations ('fc7') as features for all the images using AlexNet [10] architecture pre-trained on HybridCNN. Each feature is 4096-dimensional. HybridCNN is a CNN model pretrained on 978 object categories from ImageNet database [1] and 205 scene categories from Places dataset [34].

Then we use our trained event concept classifiers to predict the concept score for each image. We concatenate the concept scores to form the final feature vector for each image. These scores are then input to a linear SVM (We use the publicly available LIBLINEAR library [70]) that trains a sentiment detection model for each sentiment class and predicts the sentiment of the 30% test samples per class. We evaluate the effectiveness of our algorithm by computing the sentiment prediction accuracy for each class and the overall average accuracy.

3.5 Results and Discussion

Table 6.2 shows the sentiment prediction accuracies for several powerful state-of-the-art baselines and our proposed event concept features on our event sentiment dataset. We use the SentiBank [73] and Deep SentiBank [92] implementations provided by the authors. We also compare against the baselines of directly using fc7 features from AlexNet [10] and HybridCNN and training a sentiment classifier on top of the fc7 features. For all the sentiment classes as well as overall average sentiment prediction, our proposed approach outperforms the state-of-the-art. This is achieved given that our method does not use sentiment-specific concepts such as 'smiling baby'. Our method also shows superior performance to deep CNN features (AlexNet and HybridCNN), demonstrating that off-the-shelf deep CNN features are insufficient to recognize sentiments in event images containing crowded and complex scenes.

The reason why sentiment-specific mid-level representation (adjective noun-pairs) does not work well with social event images is that concepts such as 'magical sunset' or 'amazing sky' may be relevant for general images shared on the web but social event images comprise complex interplay of objects, people and scenes. Our event concepts such as 'shouting slogans' or 'birthday girl' are event specific and generalize to many different events.

Sample positive and negative images correctly classified by our proposed method are shown in Figure 3.6. The positive images (first row) have the following event concepts predicted on them: 'crowd parade', 'troupe performs', 'party students', 'streets' etc. The second row depicts negative sentiment images that are correctly identified. It is apparent that the colors in the image also affect the sentiment annotation and thus we see dark black and gray tones in some of the negative images. Sample negative images with their top predicted concepts are shown in Figure 3.7.

However, there are some event images where our sentiment classifier does not predict

Table 3.1: Per-class and average accuracy (in %) of event image sentiment prediction.

Features	positive	negative	neutral	avg. accuracy
AlexNet CNN	64.67	35.25	63.96	54.63
Hybrid CNN	72.15	67.08	61.27	66.83
SentiBank	62.31	60.79	59.09	60.73
Deep SentiBank	74.52	71.74	65.83	70.69
Event concepts (ours)	**77.11**	**74.13**	**67.94**	**73.06**

Figure 3.6: Correct positive (top row) and negative (bottom row) sentiment predictions by our proposed method on the social event dataset

Figure 3.7: Top predicted concepts for sample negative images in our dataset

the correct sentiment. This is due to the subjectivity in deciding which image evokes a neutral or negative emotion as can be seen in Figure 3.8. Since there are images in these color tones in the dataset which are labeled as negative, the classifier predicted negative

sentiment on these images.

Figure 3.8: Neutral sentiment images but classifier predicts them as negative images

Figure 3.9: Neutral sentiment images but classifier predicts them as positive images

Similarly there are images annotated as 'neutral' but the classifier predicts them as positive due to the stronger positive cues present in these images as depicted in Figure 3.9. A possible solution to this is to add more training data explicitly drawing the line between positive and neutral sentiment and negative and neutral sentiment in complex event images. It constitutes a promising direction for future extensions of this work.

Table 3.2: Top predicted concepts for positive, negative and neutral images on characterization dataset.

Sentiment	**Top predicted concepts**
Positive	concert, festivities, party, birthday celebrations, food, wedding church, bride heart, homecoming parade
Negative	protesting, politics protest, police parade, riots, parading, marchers protest, antiwar demonstrations
Neutral	horribles parade, diploma, rally, activism, house concert, street, paint balling, party students, fall graduates

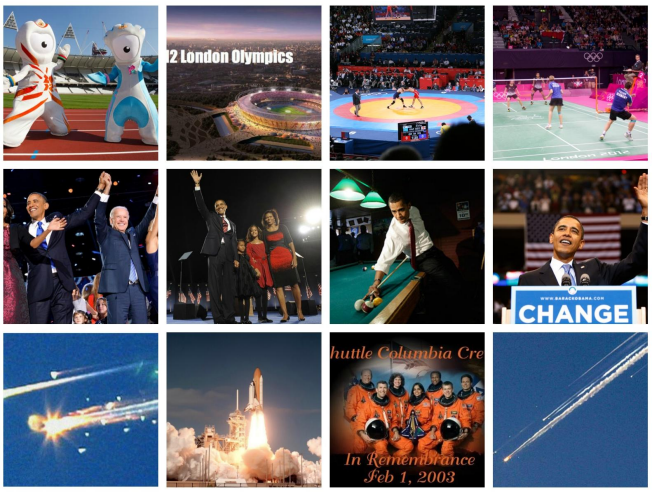

Figure 3.10: Sample images from the characterization dataset used for qualitative analysis. From top to bottom, the events are: *Summer Olympics 2012, Obama wins elections 2008 and Columbia Space Shuttle Disaster*

CHAPTER 4

VIDEO ACTIVITY RECOGNITION WITH MINIMAL SUPERVISION

In this chapter, we address the problem of semi-supervised video action recognition using Generative Adversarial Networks as an intermediate representation. Our proposed framework involves training a deep convolutional generative adversarial network (DCGAN) using a large video activity dataset without label information. Then we use the trained discriminator from the GAN model as an unsupervised pre-training step and fine-tune it on a labeled dataset to recognize human activities. We determine good network architectural and hyperparameter settings for using the discriminator from DCGAN as a trained model to learn useful representations for action recognition. Our preliminary results demonstrate that semi-supervised framework using only appearance information achieves superior or comparable performance to the current state-of-the-art semi-supervised action recognition methods on two challenging video activity datasets: UCF101 and HMDB51.

4.1 Introduction

One of the biggest challenges in recognizing activities in videos is obtaining large labeled video datasets. Annotating videos is largely both expensive and cumbersome due to variations in viewpoint, scale and appearance within a video. This suggests a need for semi-supervised approaches to recognize actions in videos. One such approach is to use deep networks to learn a feature representation of videos without activity labels but with temporal order of frames as a 'weak supervision' [102, 103]. This approach still requires some supervision in terms of deciding sampling strategies and related video encoding methods to input to neural networks (such as dynamic images [104]) and designing 'good questions' of correct/incorrect orders as input to the deep network.

Generative models such as the recently introduced Generative Adversarial Networks

(GANs) [26] approximate high dimensional probability distributions like those of natural images using an adversarial process without requiring expensive labeling. To this end, our research question is: *How can we use abundant video data without labels to train a generative model such as a GAN and use it to learn action representation in videos with little to no supervision?*

GANs are conventionally used to learn a data distribution of images starting from random noise. Adversarial learning in GANs involves two networks: a discriminator network and a generator network. The discriminator network is trained on two kinds of inputs – one consisting of samples drawn from a high dimensional data source such as images and the other consisting of random noise. Its goal is to distinguish between real and generated samples. The generator network uses the output of a discriminator to generate 'better' samples. This minimax game aims to converge to a setting where the discriminator is unable to distinguish between real and generated samples. We propose to use the discriminator trained to only differentiate between a real and generated sample for learning a feature representation of actions in videos.

We use the GAN setup to train a discriminator network and use the learned representation of discriminator as "initialized weight." Then fine-tune that discriminator on labeled video dataset such as UCF101 [32]. Recent works have done small experiments [105] but to our knowledge, nobody has done an in-depth study and especially considered all the architecture/hyperparameter settings that can give you a good performance across datasets (we do well on HMDB51 too) using only appearance information in the video. This unsupervised pre-training step avoids any manual feature engineering, video frame encoding, searching for the best video frame sampling technique and results in an action recognition performance competitive to the state-of-the-art using only appearance information. Furthermore, since the approach involves 64 x 64 sized inputs, it is applicable to real world low resolution videos.

Our key contributions and findings are:

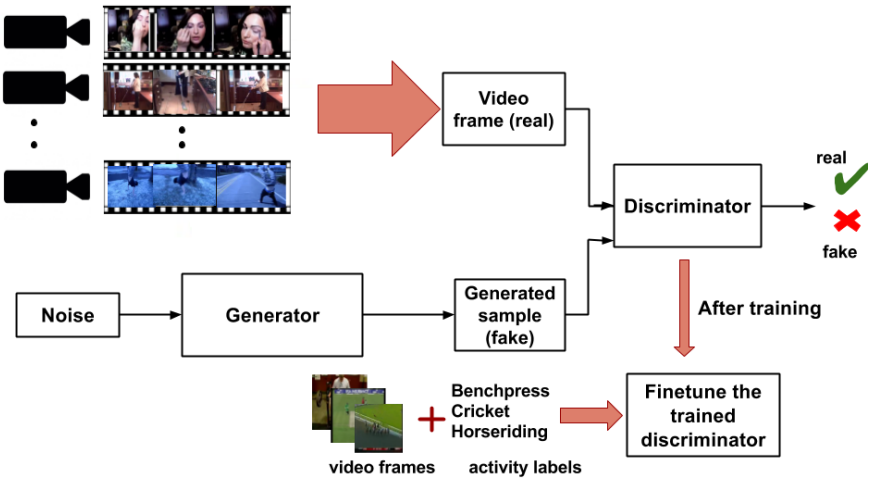

Figure 4.1: Our approach to learn action representation from GANs

- We propose a systematic semi-supervised approach to learn action representations from videos using GANs.
- We perform a comprehensive study of best practices to recognize actions from videos using the GAN training process as a good initialization step for recognition.
- We find that appearance-based unsupervised pre-training for video action recognition performs superior or comparable to the state-of-the-art semi-supervised multi-stream video action recognition approaches.
- Our method is applicable to very low resolution videos since we work with 64 x 64 sized inputs.
- Our unsupervised pre-training step does not require weak supervision or computationally expensive steps in the form of video frame encoding, video stabilization and search for best sampling strategies.

4.2 Related Work

To date, action recognition is one problem in Computer Vision where deep Convolutional Neural Networks (CNNs) have not outperformed hand-crafted features. Action recognition from videos has come a long way from holistic feature learning such as Motion Energy Image (MEI) and Motion History Image (MHI) [106], space-time volumes [107] and Action Banks [108] to local feature learning approaches such as space-time interest points [109], HOG3D [110], histogram of optical flow [111] and tracking feature trajectories [112, 113, 114, 115].

The recent success of CNNs in image recognition has enabled many researchers to treat a video as a set of RGB images, perform image classification on the video frames and aggregate the network predictions to achieve video level classification [116]. Our approach is also inspired by local appearance encoding methods for videos. 3D convolutional networks capture spatio-temporal features via 3D convolutions in both spatial and temporal domains [117]. Various fusion techniques are proposed to pool the temporal information to construct video descriptors [4, 118]. Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks have also been used to model videos for action recognition [119, 120]. Using multiple networks to model appearance and motion was first introduced by Simonyan and Zisserman [116]: the two-stream architecture, where the spatial architecture is the standard VGG Net [121] and the temporal stream network takes input stacked optical flow fields. Wu *et al.* [122] added audio and LSTMs to the network to improve video classification performance. We do not experiment with multiple modalities in this paper as we use only RGB frames as input to the model for our proof of concept.

Generative models have been successfully used to avoid manual supervision in labeling videos with the most common application being video frame prediction [123, 124, 125, 126, 127, 128, 125, 129, 130]. Since appearance changes are smooth across videos, temporal consistency [131] and other constraints [132] are useful to learn video representations.

Our work proposes a generative model as an unsupervised pre-training method for action recognition. While approaches that take temporal coherency into account such as [102, 126, 30, 133] are similar to our work, they are different in that enforcing temporal coherency still involves weak supervision [102] where they have to pre-select good samples from a video. We do not do any weak supervision in our approach but only use the generative adversarial training as an unsupervised pre-training step to recognize actions.

Recently [103] train a network to predict the odd video out of a set of videos where the "odd one out' is a video with its frames in wrong temporal order. The key difference between our work and theirs is that we do not require any weak supervision in terms of selecting the right video encoding method, sampling strategies or designing effective odd-one-out questions to improve accuracy.

Generative Adversarial Networks [26] have been used for semi-supervised feature learning particularly after the introduction of Deep Convolutional GANs (or DCGANs) [134]. Radford [134] *et al.* use the discriminator (pre-trained on ImageNet) to compute features on CIFAR10 dataset [135] for classification. Other works to use GANs for semi-supervised learning [136, 137, 138, 139, 140] are all designed for image recognition, not videos.

A recent work is [105] where the authors train GANs for tiny video generation. They fine-tune their trained discriminator model on UCF101 and show promising results. However, their model is significantly more complicated and requires stabilized videos which involves SIFT [141] and RANSAC [142] computation per video frame, something that is not required by our method.

4.3 Approach

We briefly review the main principles behind GAN models and describe our methodology in detail to recognize actions by leveraging their unsupervised feature learning capability on videos.

4.3.1 Generative Adversarial Networks

GAN networks [26] exploit game theoretic approaches to train two different networks; a generator and a discriminator. The generator represented by function G parameterized by $\theta^{(G)}$ starts with an input noise vector z that is sampled from a normal distribution $p_{noise}(z)$, up-samples this noise distribution and outputs an image $\hat{I}$. The discriminator network is a CNN network (represented by function D) parameterized by $\theta^{(D)}$ that takes as input an image (I (real image) or $\hat{I}$ (generated or fake image)) and outputs a probability $\in \{0, 1\}$ that whether the input image is from the real distribution or generated distribution. Training GANs involve a minimax game in which the generator attempts to 'fool' the discriminator into predicting a generated image as real whereas the discriminator attempts to identify correctly which input images are fake. The discriminator cost function is a cross entropy loss defined by:

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = \mathbb{E}_{I \sim p_{data(I)}}[logD(I)] + \mathbb{E}_{z \sim p_{noise(z)}}[\log(1 - D(G(z)))] \tag{4.1}$$

The minimax objective from Equation 4.1 can be optimized using gradient-based methods since both discriminator and generator are functions (D and G) that are differentiable with respect to their inputs and parameters [143]. The solution to this problem is a Nash equilibrium as both functions are trained to minimize their costs while maximizing the other's objective. GANs can be trained using Stochastic Gradient Descent (SGD) with any optimizer of choice.

4.3.2 Training GANs with Video Frames

So far in the research community, GANs have been primarily used for sample generation. Thus, focus has been on modifying generator parameters and loss functions in order to generate higher resolution images with minimal artifacts. The discriminator network in

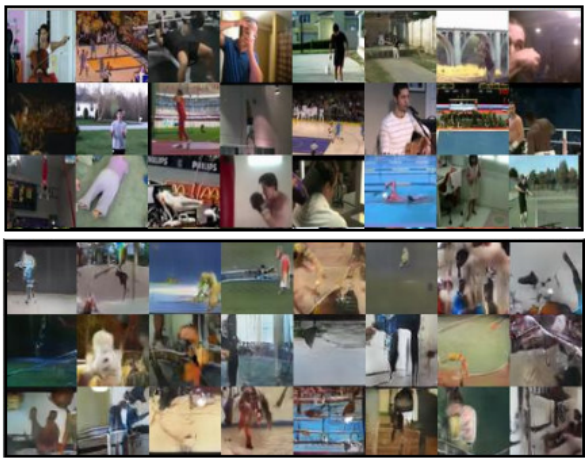

Figure 4.2: Results after 100 epochs of running DCGAN [134] on UCF101 video frames. The images in the top three rows are real while those on the bottom are generated by the model

all variants of GANs is trained with binary cross entropy loss (see Equation 4.1) [143]. Since our focus is not image generation but learning useful features to transfer to the task of action recognition, we are motivated to train and use the discriminator network in GANs for action recognition. The discriminator network in a GAN learns a representation of local appearance features thus modeling objects and scenes in video frames as context. Lastly, it does so in an unsupervised manner i.e. we do not require explicit labels for objects, scenes or actions to pre-train our action recognition model.

Consider a set of videos $\mathcal{V}$ where $\mathcal{V} = \{V_1, ..., V_n\}$ and n is the number of videos in the dataset. Each video consists of a variable number of frames (sampled at the rate of one frame per second). We use all the frames in the training set of videos from two challenging video activity datasets without any label information to train the GAN model. Our approach is shown in Figure 4.1. We train GANs using a variety of techniques proposed in prior research to generate images. To compare with GANs pre-trained on an object recognition dataset, we also train a GAN model on ImageNet [1] images. We use the same architecture as proposed in the DCGAN [134] paper since the authors have demonstrated the transfer learning capability of DCGAN model on CIFAR10 dataset.

4.3.3 Unsupervised Pre-training

When dealing with small datasets, a CNN's generalization performance decreases so that the test accuracy remains small even while training accuracy may increase. This is why a common practice is to initialize the weights of the layers with ImageNet pre-trained CNN weights instead of training from scratch. This is referred to as supervised pre-training since ImageNet labels have been used to determine the initial weights.

Our approach is different in that we are trying to do **unsupervised** pre-training - determining starting weights for a CNN model (discriminator) which is pre-trained without label information using adversarial training. This unsupervised pre-training setup is compared with initializing the weights in the discriminator network using other settings and we show that the GAN-based initialization significantly outperforms other initialization strategies on the test set of UCF101.

4.3.4 Fine-tuning Discriminator Model

In this step of our approach we initialize the network with the learnt weights from adversarial training and fine-tune it on two video activity datasets. In the process of fine-tuning, we are faced with numerous choices of network architecture, learning rate schemes, optimization and data augmentation. We explore in the space of these variations and report all results on the test split 1 of UCF101 dataset.

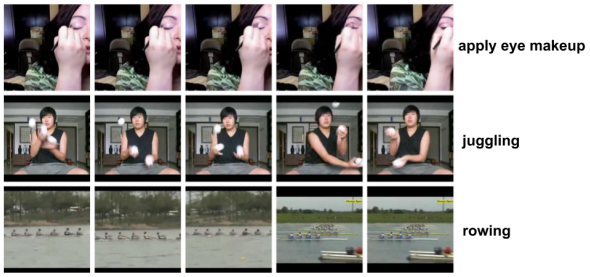

Figure 4.3: Sample frames from the UCF101 dataset [32] with action classes (from top to bottom): apply eye makeup, juggling balls and rowing

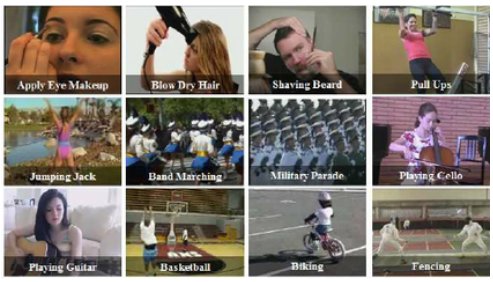

Figure 4.4: Sample frames from the HMDB51 dataset [144]

4.4 Experiments

4.4.1 Datasets

UCF101 [32] is a benchmark action recgonition dataset comprising 13320 YouTube videos of 101 action categories. Actions include human-object interactions such as 'apply lipstick', body motion such as 'handstand walking', human-human interactions, playing musical instruments and sports. The dataset is small but challenging in that the videos vary in viewpoint changes, illumination, camera motion and blur. The second dataset we experiment on is the HMDB51 dataset [144] which contains 6766 videos of 51 actions such as chew, eat, laugh etc.. Sample frames from both datasets are shown in Figures 4.3 and 4.4.

4.4.2 Unsupervised Pre-training

This section describes three experiments to determine: (a) Whether GANs can generate action images (b) Training Protocol of GANs and (c) Data Augmentation steps

Can GANs Generate Action Images? Since we consider a video as a set of RGB frames, we address the first question: Are GANs, traditionally used for generating faces, objects and scenes capable of generating an image representing an action? This question is crucial to address because it determines the validity of using the trained GAN discriminator as a CNN network and fine-tune it on a labeled video activity dataset. To answer this question, we use all the videos from the train split 1 of UCF101 [32] and sample 1 frame per second

from each video. We train a DCGAN model with default parameters and after 100 epochs, obtain results shown in Figure 4.2. From visual inspection we can see that vanilla DCGAN is able to learn a coarse representation of activities involving humans. The question now remains whether we can use the feature representation learned by GAN's discriminator as an unsupervised pre-training step to classify actions in labeled video action recognition datasets.

Training Protocol of GANs: We use DCGAN's public implementation in torch and train three separate GAN models: One with UCF101 video frames, second with ImageNet [145] images and third with a subset of Sports1M dataset [146] frames. We train all three models for 100 epochs using the architectural guidelines proposed in [134], namely, batch normalization [147] in discriminator as well as the generator, leaky Rectified Linear Units (leaky ReLU) [148] in all layers of discriminator, strided convolutions in discriminator instead of pooling layers and fractional-strided convolutions in the generator. There are no fully-connected (FC) layers in the DCGAN architecture as the authors of [134] report no loss in generator performance for not including FC layers. Hence we also use the same architecture for training the GAN model.

Data Augmentation: The main difference between our GAN training and the DCGAN [134] approach is that DCGAN [134] performs data augmentation via taking 64 x 64 sized random crops of the image as well as scaling the images to range [-1,1]. This scaling is done for the tanh activation function in the generator. We change that protocol and avoid random cropping. We only scale the frames of videos to the range [-1,1] and scale the size to 64 x 64. The reason why we avoid random cropping is because the action frames from videos are much larger and contain much more information than the original images used for training DCGAN (bedrooms, faces and the like). Taking random crops from action frames will not result in a useful representation because too much information will be lost. Thus, we only scale the images to 64x64 as our aim is not just to generate action images but to learn an effective action representation for recognition.

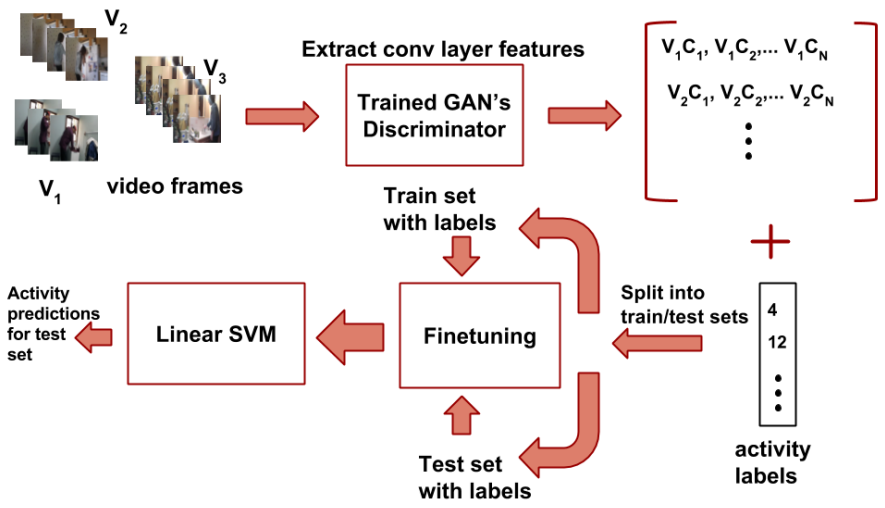

Figure 4.5: Our approach: From training GANs to classifying actions in videos

4.4.3 Fine-tuning for Action Recognition

Here we describe the set of experiments conducted after the GAN model has been trained. We use the pre-trained discriminator network from our GAN model and fine-tune it on the two labeled video action datasets: UCF101 [32] and HMDB51 [144]. We begin by replacing the last spatial convolutional layer (CONV5) with one that has the correct number of outputs (equal to the number of action classes). See Figure 4.6. This layer is initialized randomly and the network is trained again with the previous layers initialized with the pre-trained discriminator's weights.

We perform a comprehensive experimental analysis of architectural choices, hyperparameter settings and other good practices and report the accuracy on the test set of UCF101 dataset.

Does Source Data Distribution Matter? In this experiment, we determine whether the dataset we train GAN with (which we refer to as the *source dataset*) determines performance on the *target dataset* (the labeled dataset on which we fine-tune the discriminator network). To this end, we train DCGAN on three large scale datasets: ImageNet [1] images, UCF101 [32] video frames and frames of 10,000 videos from Sports1M [4] dataset.

Table 4.1: Comparing the accuracy on target dataset with three large scale datasets used to train GAN models

	Destination Dataset (accuracy %)	
Source Dataset	**UCF101**	**HMDB51**
ImageNet	43.88	12.82
UCF101	**47.20**	12.94
Sports1M	42.50	**13.02**

We use the same sampling strategy of 1 frame per second for both video datasets and train all three GAN models separately for 100 epochs.

Our experimental setup is shown in Figure 4.5. For each video V_i, there is a set of frames F_i where $F_i = \mathcal{V} = \{f_{n_1}, f_{n_2}, ..., f_{n_i}\}$ where n_i is the number of frames extracted for video V_i. Each video's frames are passed through the trained GAN's discriminator and we extract CONV4's activations as features on each frame. We average frame-level features to obtain video-level features. We train a linear SVM classifier [70] on top of these features using the train/test split1 provided by the dataset authors and obtain classification accuracy on the test set. We use the same setting for training all three GAN models as described in the training protocol earlier. Our results are shown in Table 4.1.

As can be seen from Table 4.1 training a GAN with UCF101 frames results in the best test accuracy on both UCF101 and HMDB51. The difference between training a GAN model with ImageNet and Sports1M frames and training it with UCF101 frames is significant. Note that we did not use all videos from the Sports1M dataset; we randomly selected 10,000 videos from the dataset, extracted 1 frame per second from each video and used those frames to train the GAN model. For HMDB51 dataset the difference in test accuracy between using a GAN discriminator pre-trained on UCF101 and other datasets is not very large. But the superior performance of training a GAN model with video action frames is clearly demonstrated by this experiment. The features learned by the discriminator network are strong enough to transfer to other video datasets as well.

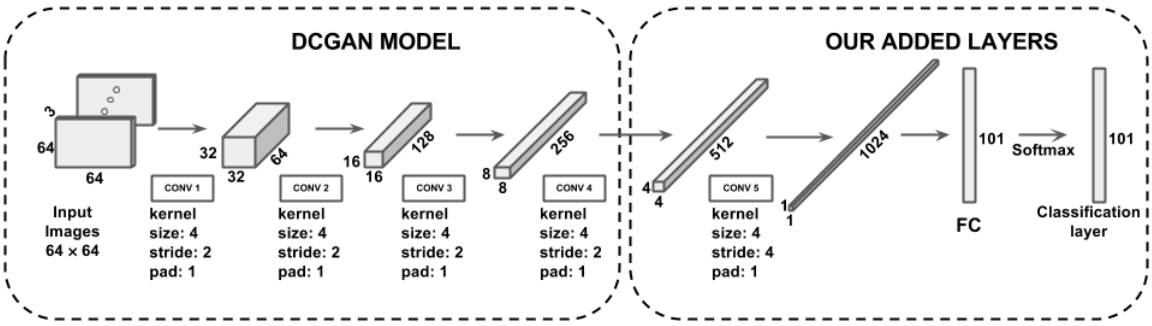

Figure 4.6: Our network architecture: DCGAN discriminator architecture on the left and our added layers on the right

Choice of Architecture: There are several ways of changing the architecture of the pre-trained discriminator network for fine-tuning. Note that the discriminator is just another CNN network with spatial convolutional layers and no fully connected layers. For fine-tuning on the UCF101 dataset, we replace the last convolutional layer (CONV4) with one that has the correct number of outputs, initialize this layer randomly and train this network (fine-tune) for 160 epochs. This fine-tuning experiment is called 'CONV4' in Table 4.2. Network depth determines the model's performance both in theory and practice [149]. Hence we add another convolutional layer (CONV5) and a fully connected layer (FC), initialize them from scratch and retrain the network till convergence. We extract CONV4, CONV5 and FC features from the finetuned network. We concatenate CONV5 and CONV5 features and test the performance as well as CONV4, CONV5 and FC features. We do not freeze any layers before fine-tuning and keep a learning rate of 0.001 to fine-tune the network. We empirically found that freezing the earlier layers and finetuning only the last layer(s) did not increase performance. We train a linear SVM on top of the extracted features and compute results on UCF101's test set. Our results are shown in Table 4.2. Our network architecture is shown in Figure 4.6.

Our results in Table 4.2 show that with all other parameters kept the same, adding a convolutional layer and a fully connected layer in the discriminator network architecture results in only a slight improvement in performance. We note that this is not a huge dif-

ference and this may seem counterintuitive but the reason why this happens is that we are initializing the added network layers randomly before fine-tuning. Also, the dataset size of UCF101 frames is not very large with 84,747 frames in the training set and 33,187 frames in the test set. This may lead to over fitting resulting in only a slight increase in performance on the test set especially when the fully connected layer is added.

To reduce overfitting, we add dropout [150] after the additional convolutional and fully connected layers. We note the performance with/without dropout by extracting CONV4 features from both networks (after finetuning) and training a linear SVM. Our results are shown in Table 4.3. Adding dropout regularizes the network more thus increasing the performance on test set of UCF101.

Fine-tuning vs Linear SVM: Once we fine-tune the discriminator model on the datasets, we have a choice of whether to extract the CONV4's activations and train a linear SVM on top of it or fine-tune the last layers with softmax classifier. We do both in our experiments and note that the outcome is dependent on the dataset. We find that when we fine-tune the discriminator network on UCF101, the test set accuracy using softmax is lower than extracting CONV4 features and training a linear SVM to recognize actions. However when using HMDB51, the softmax classification on the test set results in a higher accuracy than extracting Layer 9 features and training a linear SVM classifier. This result is shown in Table 4.4.

From Table 4.4 it is apparent that for UCF101, feature embedding and training a linear SVM results in a better accuracy than softmax classification. The complete opposite is true with HMDB51 dataset. We dig deeper to investigate why this happens. We find that the

Table 4.2: Effect of making the network deeper: Adding more layers slightly improves action recognition performance

Architectural changes	**Test Accuracy** (%)
CONV4	48.35
CONV4 + CONV5 + FC	49.30
CONV4 + CONV5	50.12

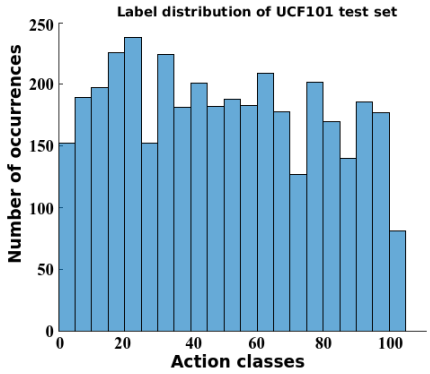

Figure 4.7: Label distributions of UCF101 test set. The HMDB51 dataset has uniform distribution of 30 videos per action class

label distribution of the dataset on which a deep network is being fine-tuned on is the key to determine which method results in a better test accuracy. The label distribution of UCF101 test set is shown in Figure 4.7. This distribution is not balanced while that of HMDB51 is completely balanced in terms of number of videos per action category. Hence it appears that when classes are unbalanced, since we have not used weighted loss in the neural network, the linear SVM learns the features better hence resulting in an increased performance on the test set. In the case of HMDB51, all classes are balanced equally leading to the superior performance of the softmax classifier over the feature embedding approach.

Unsupervised Pretraining vs Random Initialization We validate the use of our unsupervised pre-training approach by comparing it with a network that is initialized randomly. We initialize all the layers of the network using 'xavier' initialization. Proposed by [151], the authors recommend initializing weights by drawing from a distribution with zero mean and variance given by: $Var(W) = 2/(n_{in} + n_{out})$ where W is the distribution which the

Table 4.3: Effect of adding dropout: Adding dropout layers improves action recognition performance

Architectural changes	**Test Accuracy** (%)
CONV4 (with dropout)	48.35
CONV4 (without dropout)	45.68

Table 4.4: Comparing two ways of evaluating fine-tuned network performance on UCF101 and HMDB51 test sets

	Accuracy (%) on test set	
	CONV4 + linear SVM	Softmax
UCF101	**48.35**	41.40
HMDB51	14.40	**21.04**

Table 4.5: Results on UCF101 test set with network initialized with our unsupervised pre-trained weights vs initialized using the method of [151]

Initialization	**UCF101 (%)**	**HMDB51 (%)**
Xavier + finetuning	33.10	11.6
DiscrimNet (ours) + finetuning	**49.30**	**20.4**

neuron is initialized with, n_{in} is the number of neurons feeding into the layer and n_{out} is the number of output neurons from this layer. We initialize all layers with this scheme and train the network till convergence on UCF101. For HMDB51, we train a network for 50 epochs with xavier initialized layers and compare that to our proposed discriminator initialized method at 50 epochs. The results are shown in Table 4.5 and clearly validate the use of our unsupervised pretraining approach to initialize the network before finetuning.

4.4.4 Preliminary Results

We compare our approach with several recent semi-supervised baselines which recognize actions in videos. The baselines are:

- *STIP features*: Handcrafted Space Time Interest Point (STIP) features introduced by [109].
- *DrLim [152]*: This method uses temporal coherency by minimizing the L2 distance metric between features of neighboring frames in videos and enforcing a margin δ between far apart frames.
- *TempCoh [129]*: Enforce temporal coherence by using L1 distance instead of L2. Similar to DrLim [152].

Table 4.6: Comparing our method to state-of-the-art semi-supervised approaches on UCF101

Method	**UCF101-split1 (%)**
STIP features [111]	43.9
DrLim [152]	45.7
TempCoh [129]	45.4
Obj. Patch [30]	40.7
Shuffle [102]	50.9
VideoGAN [105]	52.1
O3N [103]	60.3
DiscrimNet (ours) CONV4 + linear SVM	49.33
DiscrimNet (ours) CONV5 + linear SVM	48.88
DiscrimNet (ours) (CONV4 + CONV5) + linear SVM	50.12

Table 4.7: Comparing our method to state-of-the-art semi-supervised approaches on HMDB51

Method	**HMDB51** (%)
DrLim [152]	16.3
TempCoh [129]	15.9
Obj. Patch [30]	15.6
Shuffle [102]	19.8
O3N [103]	32.5
DiscrimNet (ours) (fine-tuned)	21.0

- *Obj. Patch [30]*: They extract similar object patches using videos and learn a representation of objects by tracking them through time. This model is used and fine-tuned on UCF101 by [102].
- *Shuffle [102]*: They use sequence verification as an unsupervised pre-training step for vidoes. The model is then fine-tuned on UCF101.
- *VideoGAN [105]*: They generate tiny videos using a two stream GAN network. Their model is fine-tuned on UCF101.
- *O3N [103]*: They use odd-one-out networks to predict the wrong temporal order from the right ones. Their model is then fine-tuned on UCF101.

The results are shown in Table 4.6 and Table 4.7.

4.5 Discussion

Our comparison with several state-of-the-art semi-supervised approaches to recognize actions in vidoes yields important insights. Our results show competitive performance as compared to the state-of-the-art approaches in semi-supervised learning given that:

- We only use appearance features and do not experiment with motion content of the video. This is especially intriguing given that our method outperforms STIP features on this dataset. All methods in the results we compare to use temporal coherency as a signal and do motion encoding.
- We do not do weak supervision in the form of temporal consistency and do not design temporal order based networks. The only supervision provided to the GAN is the difference between a real image and noise.
- Our model outperforms several state-of-the-art approaches on HMDB51 given that no video from the dataset was used in the unsupervised pre-training step of this approach. This shows the domain adaptation capability of GAN discriminator networks and that they are able to capture enough information to learn useful representation of actions in video frames.

The methods that outperform our proposed approach are either computationally expensive or require much more supervision in the form of selecting sampling strategies, video encoding methods or in the case of O3N networks [103], designing effective odd-one-out questions for the network to learn feature representations for action recognition.

4.6 Conclusion

In this chapter we propose an unsupervised pre-training method using GANs for action recognition in videos. Our method does not require weak supervision in the form of temporal coherency, sampling selection or video encoding methods. Purely on appearance

information alone, our method performs either better than or comparable to the state-of-the-art semi-supervised action recognition methods.

CHAPTER 5

VIDEO JIGSAW: UNSUPERVISED LEARNING OF SPATIOTEMPORAL CONTEXT FOR VIDEO ACTION RECOGNITION

In the previous chapter, we proposed an appearance-based semi-supervised approach to categorize activities in videos using generative adversarial networks (GANs). Our best model achieved comparable accuracies to the state-of-the-art. The next question we address in this chapter is: How to incorporate temporal dynamics in the unsupervised pretraining step such that it gives us a better result on activity recognition. Therefore, we propose a self-supervised learning method to jointly reason about spatial and temporal context for video recognition. Recent self-supervised approaches have used spatial context [153, 154] as well as temporal coherency [102] but a combination of the two requires extensive preprocessing such as tracking objects through millions of video frames [155] or computing optical flow to determine frame regions with high motion [156]. We propose to combine spatial and temporal context in one self-supervised framework without any heavy preprocessing. We divide multiple video frames into grids of patches and train a network to solve jigsaw puzzles on these patches from multiple frames. So the network is trained to correctly identify the position of a patch within a video frame as well as the position of a patch over time. We also propose a novel permutation strategy that outperforms random permutations while significantly reducing computational and memory constraints. We use our trained network for transfer learning tasks such as video activity recognition and demonstrate the strength of our approach on two benchmark video action recognition datasets without using a single frame from these datasets for unsupervised pretraining of our proposed video jigsaw network.

Figure 5.1: Video Jigsaw Task: The first row shows a tuple of frames of action "high jump". Second row shows how we divide each frame into a 2x2 grid of patches. The third row shows a random permutation of the 12 patches which are input to the network. The final row shows the jigsaw puzzle assembled

5.1 Introduction

Self-supervised tasks that exploit spatial context include predicting the location of one patch relative to another [153], solving a jigsaw puzzle of image patches [154], predicting an image's color channels from grayscale [157, 158] among others. Self-supervision tasks on video data include video frame tuple order verification [102], sorting video frames [156] and tracking objects over time and training a Siamese network for similarity based learning [30]. Video data involves not just spatial context but also rich temporal structure in an image sequence. Attempts to combine the two have resulted in multi-task learning approaches [159] that result in some improvement over a single network. This work proposes a self-supervised task that jointly exploits spatial and temporal context in videos by dividing multiple video frames into patches and shuffling them into a jigsaw puzzle problem. The network is trained to solve this puzzle that involves reasoning over space and time.

Figure 6.1 shows our proposed approach, which we call *video jigsaw*. Our contributions in this work are:

1. We propose a novel self-supervised task which divides multiple video frames into

patches, creates jigsaw puzzles out of these patches and the network is trained to solve this task.

2. Our work exploits both spatial and temporal context in one joint framework without requiring explicit object tracking in videos or optical flow based patch mining from video frames.

3. We propose a permutation strategy that constrains the sampled permutations and outperforms random permutations while being memory efficient.

4. We show via extensive experimental evaluation the feasibility and effectiveness of our approach on video action recognition.

5. We demonstrate the domain transfer capability of our proposed video jigsaw networks, given that our best self-supervised model is trained on Kinetics [31] videos and we demonstrate competitive results on UCF101 [32] and HMDB51[33] datasets.

5.2 Related Work

Unsupervised representation learning is a well studied problem in the literature for both images and videos. The goal is to learn a representation that is *simpler* in some way: it can be low-dimensional, sparse, and/or independent [160]. One way to learn such a representation is to use a reconstruction objective. Autoencoders [161] are neural networks designed to reconstruct the input and produce it as its output. Denoising autoencoders [162] train a network to undo random corruption of the input data. Other methods that use reconstruction to estimate the latent variables that can explain the observed data include Deep Boltzmann Machines [163], stacked autoencoders [164, 165] and Restricted Boltzmann Machines (RBMs) [166, 167]. Classical work (before deep learning) involved hand-designing features and feature aggregation for application such as object discovery in large datasets [168, 169] and mid-level feature mining [170, 171, 172].

Unsupervised learning from videos include many learning variants such as video frame prediction [29, 129, 127, 130, 126] but we argue that predicting pixels is a much harder task, especially if the end task is to learn high level motion and appearance changes in frames for activity recognition. Other unsupervised representation learning approaches include exemplar CNNs [173], CliqueCNNs [174] and unsupervised similarity learning by clustering [175].

Unsupervised representations are generally learned to make another learning task (of interest) easier [160]. This forms the basis of another line of work that has emerged, called 'self-supervised learning' [153, 102, 157, 158, 30, 159, 155, 176]. Self-supervised learning aims to find structure in the unlabeled data by designing auxiliary tasks and pseudo labels to learn features that can explain the factors of variation in the data. These features can then be useful for the target task; in our case, video action recognition. Self-supervised learning can exploit several cues, some of which are spatial context and temporal coherency. Other self-supervised learning tasks on videos use cues like ego-motion [124, 177, 178] as a supervisory signal and other modalities beyond raw pixels such as audio [179, 180] and robot motion [181, 182, 183, 184]. We briefly cover relevant literature from the spatial, temporal and combined contextual cues for self-supervised learning.

Spatial Context: These methods typically sample patches from images or videos. Supervised tasks are designed around the arrangement of these patches and pseudo labels constructed. Doersch *et al.* [153] divide an image into a 3x3 grid, sample two patches from an image and train a network to predict the location of the second patch relative to the first. This prediction task requires no labels but learns an effective image representation. Noroozi and Favaro [154] also divide an image into a 3x3 grid but they input all patches in a Siamese-like network where the patches are shuffled and the task is to solve this jigsaw puzzle task. They report that with just 100 permutations, their network is able to learn a representation such that when finetuned on PASCAL VOC 2007 [185] for object detection and classification, it produces good results. Pathak *et al.* [186] devise an inpainting auxil-

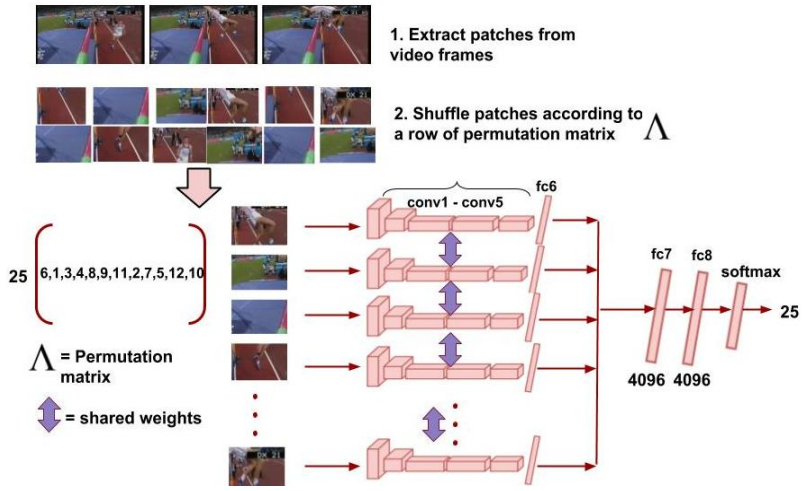

Figure 5.2: Our full video jigsaw network training pipeline.

iary task where blocks of pixels from an image are removed and the task is to predict the missing pixels. A related task is the image colorization one [157, 158] where the network is trained to predict the color of the image which is available as a 'free signal' with images. Zhang *et al.* [187] modify the autoencoder architecture to predict raw data channels as their self-supervised task and use the learnt features for supervised tasks.

Temporal Coherency: These methods use temporal coherency as a supervisory signal to train models and use abundant unlabeled video data instead of just images. Wang and Gupta [30] use detection and tracking methods to extract object patches from videos and train a Siamese network with the prior that objects in nearby frames are similar whereas other random object patches are dissimilar. Misra *et al.* [102] devise a sequence verification task where tuples of video frames are shuffled and the network is trained on the binary task of discriminating between correctly ordered and shuffled frames. Fernando *et al.* [188] design a task where they take frames in correct temporal order and shuffled order, encode them and pass them as input to a network which is then trained to predict the odd encoding out of the rest; odd being the temporally shuffled one. Lee *et al.* [156] extract high motion tuples of four frames via optical flow and shuffle them. Their network learns to predict the permutation from which the frames were sampled from. Our work is highly related

to approaches that shuffle video frames and train a network to learn the permutations. A key difference between our work and Lee *et al.* [156] is that they use only a single 80 x 80 patch from a video frame and shuffle it with three other patches from different frames. We sample a grid of patches from each frame and shuffle them with other multiple patches from other frames. Instead of the binary task of tuple verification like Misra *et al.* [102], our self-supervised task is to predict the exact permutation of the patches, much like the jigsaw puzzle task of Noroozi and Favaro [154] — only on videos. Some recent approaches have used temporal coherency-based self-supervision on video sequences to model fine-grained human poses and activities [189] and animal behavior [190]. Our model is not specialized for motor skill learning like [190] and we do not require bounding boxes for humans in the video frames as in [189].

Combining Multiple Cues: Since our approach combines spatial and temporal context into a single task, it is pertinent to mention recent approaches to combine multiple supervisory cues. Doersch and Zisserman [159] combine four self-supervised tasks in a multi-task training framework. The tasks include context prediction [153], colorization [157], exemplar-based learning [191] and motion segmentation [192]. Their experiments prove that naively combining different tasks does not yield improved results. They propose a lasso regularization scheme to capture only useful features from the trained network. Our work does not require a complex model for combining the spatial and temporal context prediction tasks for self-supervised learning. Wang *et al.* [155] train a Siamese network to recognize if an object patch belongs to a similar category (but different object) or it belongs to the same object, only later in time. This work attempts to combine spatial and temporal context but requires preprocessing to discover the tracked object patches. Our work constructs the spatiotemporal task from video frames automatically without requiring graph construction or visual detection and tracking. There is also recent work on using synthetic imagery and its 'free annotations' to learn visual representations [193] by combining multiple self-supervised tasks. A related approach to ours is that of [194] where

the authors devise two tasks for the network to train on in a multi-task framework. One is spatial placement task where a network learns to identify if a an image patch overlaps with a person bounding box or not. The second task is an ordering one where a network is trained to identify the correct sequence of two frames in a Siamese network setting much like [102]. The key difference between their work and ours is that our network does not do multi-task learning and predicts a much richer set of labels (that is, the shuffled configuration of patches) as compared to binary classification.

5.3 The Video Jigsaw Puzzle Problem

We present the video jigsaw puzzle task in this section. Our goal is to create a task that not only forces a network to learn part-based appearance of complex activities but also, how those parts change over time. For this, we divide a video frame into 2×2 grid of patches. For a tuple of three video frames, this results in $3 \times (2 \times 2) = 12$ total patches per video. We number the patches from 1 to 12 and shuffle them. Note that there are $12! = 479001600$ ways to shuffle these patches. We use a small but diverse subset of these patches' permutations, selecting them based on their Hamming Distance from the previously sampled permutations [154]. We use two sampling strategies in our experiments which we will describe in more detail. The network is trained to predict the correct order of patches. Our video jigsaw task is illustrated in Figure 6.1.

5.3.1 Training Video Jigsaw Network

Our training strategy follows a line of recent works on self-supervised learning on large scale image and video datasets [154, 156]. Typically, the self-supervised task is constructed by defining pseudo labels — in our case, the permuted order of patches. Then, each patch, after undergoing preprocessing, is input to a multi-stream Siamese-like network. Each stream, up till the first fully connected layer, shares parameters and operates independently on the frame patches. After the first fully connected layer (*fc6*), the feature representations

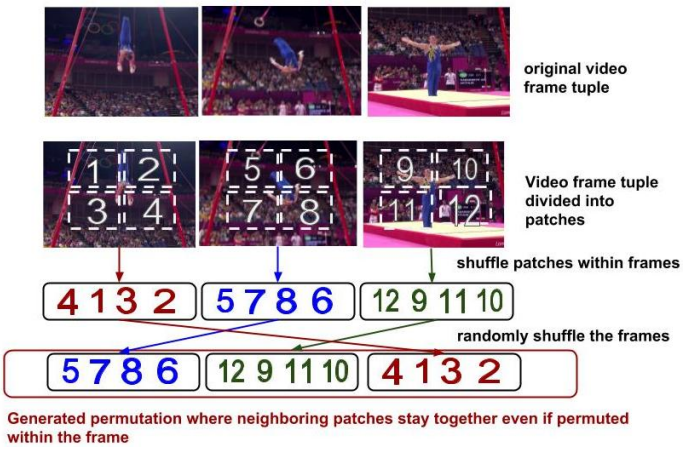

Figure 5.3: Our proposed permutation sampling strategy. We randomly permute the patches within each frame in a tuple, then we permute the frames. Since the number of patches per frame is 4, there are $4! = 24$ unique ways to shuffle these patches within a frame. We repeat this for all frames in the tuple and finally select the top N permutations based on Hamming distance. This strategy preserves spatial coherence, preserves diversity between permutations, takes a fraction of the time and memory as compared to the algorithm of [154] and results in either comparable or better performance in the transfer learning tasks

are concatenated and input to another fully connected layer (*fc7*). The final fully connected layer transforms the features to a N dimensional output, where N is the number of permutations. A softmax over this output returns the most likely permutation the frame patches were sampled from. Our detailed training network is shown in Figure 6.2.

5.3.2 Generating Video Jigsaw Puzzles

We describe here the strategy to generate puzzles from the video frame patches. Noroozi and Favaro [154] proposed to generate permutations of 9 image patches by maximizing the Hamming distance between the sampled permutations and the subsequently sampled permutations. They iterate over all possible permutations of 9 patches till they end up with N permutations; in their case, $N = 100$. In our case, since each video frame is divided into 4 patches and there are 3 frames in a tuple, it is not possible to sample permutations from all possible permutations (which is $12!$) due to memory constraints. To reimplement [154]'s approach, we devise a computationally heavy but memory-efficient means to generate 100 permutations from $12!$ possibilities. More details on how we generate these permutations

Algorithm 1: Sampling Permutations with Spatial Coherence

Input: Number of permutations N, patches per frame n_p, number of frames n_f
Output: Permutation Matrix Λ

function *generatePerm*(N, n_p, n_f)
  for $i = 1 : n_f$ **do**
    $\lambda_i \leftarrow$ random permutation of $\{n_p(i-1)+1, \ldots, n_p i\}$
    $\tilde{\lambda}^i \leftarrow$ all permutations of $\{n_p(i-1)+1, \ldots, n_p i\} : \ [\tilde{\lambda}^i_1, \ldots, \tilde{\lambda}^i_{n_p!}]^\top$
  end
  $\Lambda \leftarrow [\lambda_1^\top \ldots \lambda_{n_f}^\top]^\top$ with sub-vectors $\lambda_i^\top$ rearranged in a random order
  $F \leftarrow$ all permutations of $\{1, \ldots, n_f\} : \ [F_1, \ldots, F_{n_f!}]^\top$
  for $h = 2 : N$ **do**
    $D_{max} \leftarrow \emptyset$
    $\Lambda' \leftarrow \emptyset$
    for $f = 1 : n_f!$ **do**
      for $i = 1 : (n_p!)^{n_f - 1}$ **do**
        for $j = 2 : n_f$ **do**
          $k \leftarrow \left\lceil \frac{\left((i-1) \mod (n_p!)^{j-1}\right)+1}{(n_p!)^{j-2}} \right\rceil$
          $\tilde{\boldsymbol{\lambda}}^j \leftarrow [\tilde{\lambda}^j_k \ldots \tilde{\lambda}^j_k]^\top \in \mathbb{R}^{n_p! \times n_p}$
        end
        $\Lambda'' \leftarrow$ arrange $[\tilde{\lambda}^1 \tilde{\boldsymbol{\lambda}}^2 \ldots \tilde{\boldsymbol{\lambda}}^{n_f}]^\top$ in order F_f
        $D \leftarrow Hamming(\Lambda, \Lambda'')$
        $\overline{D} \leftarrow \frac{1}{h-1} \mathbf{1}^\top D$
        $D_{max} \leftarrow \begin{bmatrix} D_{max} & \max_k \overline{D}_k \end{bmatrix}$
        $j \leftarrow \operatorname{argmax}_k \overline{D}_k$
        $\Lambda' \leftarrow \begin{bmatrix} \Lambda' & \Lambda''_j \end{bmatrix}$
      end
    end
    $j \leftarrow \operatorname{argmax}_k D_{max(k)}$
    $\Lambda \leftarrow \begin{bmatrix} \Lambda & \Lambda'_j \end{bmatrix}$
  end
  return Λ
end

are described in Appendix A. This way, we generate the Hamming-distance based permutations as suggested by [154].

The permutation sampling approach described above treats all video frame patches as one giant image — thus, the patch belonging to the first frame may get shuffled to the last

frame's position (to maximize Hamming distance between the permutations). We treat this permutation sampling approach as an (expensive) baseline but propose another sampling strategy to minimize compute and memory constraints. Our proposed approach can scale to any number of permutations. We generate permutations with a 2×2 grid per frame. Our proposed approach forces the sampled permutations not only to obey the Hamming distance criteria but also to respect spatial coherence in video frames. This scales down computational and memory requirements dramatically while giving similar or better performance on transfer learning tasks. Our proposed permutation sampling approach is given in Algorithm 1 and visually presented in Figure 5.3.

Explanation of Algorithm 1: With the constraint of spatial coherence *i.e.* patches within a frame constrained to stay together, the full space of hashes consists of $(n_p!)^{n_f} \times n_f!$ possibilities. After generating the first hash randomly (lines 2, 3 and 5), each next hash Λ_h $h \in 2, \ldots N$ is picked by maximizing over the full space the average Hamming distance from previously generated hashes. We divide the full space into subsets of $n_p!$ hashes. Iterating through each subset Λ'' (lines 10-11), we store the best hash from the subset into Λ' along with its distance metric into D_{max} (lines 16-20). When the full space is traversed, the best from the good ones (Λ') is chosen as the new hash (lines 21-22). Lines 4, 6 and 10-15 describe how each subset Λ'' is constructed. Λ'' contains all $n_p!$ permutations of patches within the first frame but only a particular permutation of patches from the other frames. For memory efficiency, it is sufficient to only create one matrix $\tilde{\lambda}^1$ that has all patch permutations within the first frame *i.e.* it is not necessary to create $\tilde{\lambda}^i$ $i \in 2, \ldots, n_f$ as done in line 4. This is because the former is reused in every iteration but only one row from the latter is used to create $\tilde{\boldsymbol{\lambda}}^i$, the matrix of repeated rows (line 14) which can be achieved by picking the corresponding row from $\tilde{\lambda}^1$ and adding the offset $n_p(i-1)$ to each element of the row.

5.4 Experiments

In this section we describe in detail our experiments on video action recognition using the video jigsaw network and a comprehensive ablation study, justifying our design choices and conclusions. The datasets we use for training the video jigsaw network are UCF101 [32] and Kinetics [31]. The datasets we evaluate on are UCF101 [32] and HMDB51 [33] for video action recognition.

5.4.1 Datasets

UCF101 [32] is a benchmark video action recognition dataset consisting of 101 action categories and 13,320 videos; around 9.5k videos are used for training and 3.5k videos are for testing. HMDB51 [33] consists of around 7000 vidoes of 51 action categories, out of which 70% belong to training set and 30% are in the test set. Kinetics dataset [31] is a large scale human action video dataset consisting of 400 action categories and more than 400 videos per action category.

5.4.2 Video Jigsaw Network Training

Tuple Sampling Strategy For our unsupervised pretraining step on UCF101, we use the frame tuples (4 frames/tuple) provided by the authors of [156]. They extracted optical flow based regions from these frame tuples and used them in the temporal sequence sorting task [156]. We do not use the optical flow based regions from the frames but only use the tuples as a whole. For a given frame tuple f_1, f_2, f_3, f_4, we further sample three frames in the following way:
$[(f_1, f_2, f_3), (f_2, f_3, f_4), (f_1, f_3, f_4), (f_1, f_2, f_4)]$. Hence, we end up with around 900,000 frame tuples from UCF101 dataset to train our video jigsaw network on. In Kinetics dataset, each video is 10 seconds long. We create our tuples by sampling the 1^{st}, 5^{th} and 10^{th} frames from each video. The reason we do not sample further (as we did in the case

of UCF101 dataset) is simply that Kinetics dataset is very large and diverse with more than 400 videos per class. This is not true for UCF101 dataset. Note that we do not use any further preprocessing to generate the frame tuples for our video jigsaw network. Previous approaches have used expensive detection and tracking methods [30] or optical flow computation to sample the high motion patches [156].

Implementation Details We use Caffe [69] deep learning framework for all our experiments and *CaffeNet* [10] as our base network, only with 12 streams for 12 patches per tuple. Our video jigsaw puzzles are generated on the fly according to the permutation matrix Λ generated before training begins. Each row of Λ corresponds to a unique permutation of 12 patches. The video frame patches are shuffled according to the sampled permutation from Λ and input to the network. The network is trained to predict the index in Λ from which the permutation was sampled. Each video frame is cropped to 224×224, then divided into a 2×2 grid. Each grid is 112×112 pixels and we randomly sample a 64×64 patch from it. This strategy ensures that the network can not learn the location of the patches from low level appearance and texture details. We normalize each patch independently from others, to have zero mean and unit standard deviation. This is also done to prevent the network from learning low-level details (also called 'network shortcuts' in the self-supervision literature). Each patch is input to the multi-stream video jigsaw network as depicted in Figure 6.2. We use a batch size of 150 and train the network with Stochastic Gradient Descent (SGD) using an initial learning rate of 0.000128, which decreases by 10 every 128,000 iterations. Each layer in our network is initialized with xavier initialization [151]. We train the network for 500,000 iterations (approximately 80 epochs) using a Titan X GPU. Our training converges in around 62 hours.

Progressive Training Approach We borrow principles from curriculum learning [195] to train our video jigsaw network with an easy jigsaw puzzle task first and then train it for a harder task. We define an easy jigsaw puzzle task as one which has lower N as compared to a harder task as the network has to learn fewer configurations of the patches in the video

frames. So instead of starting from scratch for say, $N = 500$, we initialize the network's weights with the weights of the network with $N = 250$.

Avoiding Network Shortcuts As mentioned in recent self-supervised approaches [153, 154, 176], it is imperative to deal with the self-supervised network's tendency to learn the patch locations via low level details such as due to chromatic aberration. Typical solutions to this problem are channel swapping [156], color normalization [154], leaving a gap between sampled patches and training with a percentage of images in grayscale rather than color [176]. All these approaches aim to make the patch location learning task harder for the network. Our video jigsaw network incorporates these techniques to avoid network shortcuts. Our patch size is kept 64×64 sampled from within a 112×112 window. Around half of the total video frames are randomly projected to grayscale and we normalize each sampled patch independently. Our experiments using these techniques result in a drop in performance in video jigsaw puzzle solving accuracy but the transfer learning accuracy increases.

Choice of Video Jigsaw Training Dataset As mentioned, we train video jigsaw networks using UCF101 and Kinetics datasets. Our results using the two datasets are shown in Table 6.1. We show video jigsaw task accuracy (VJ Acc) and the finetuning accuracy on UCF101 (Finetune Acc) for pretraining with both datasets. N is the number of permutations. We can note two things from the table. Using Kinetics results in a worse video jigsaw solving performance, but results in a better generalization and transfer learning. Our finetuning results are consistently better with Kinetics pretraining as compared to training on UCF101. This shows that a large-scale diverse dataset like Kinetics is able to generalize to a completely different dataset (UCF101). One possible reason behind the reduced performance of UCF101 dataset is the fact that we oversample from it. This results in an easy task for the video jigsaw network to learn the low-level details of the video frame appearances and rapidly decrease the training loss. However, this would not result in a good transfer learning performance. To test this hypothesis, we use the reduced version of the UCF101

Table 5.1: Comparison between UCF101 and Kinetics datasets for video jigsaw training

Pretraining Dataset	**VJ Acc (%) (N = 100)**	**Finetune Acc (%) (N = 100)**	**VJ Acc (%) (N = 250)**	**Finetune Acc (%) (N = 250)**
UCF101	97.6	44.0	84.6	42.6
Kinetics	61.6	**44.6**	44.0	**49.0**

Table 5.2: Comparison between Kinetics and the original UCF101 frame tuples as pretraining dataset for video jigsaw network

Pretraining Dataset	**VJ Acc (%)**	**Finetune Ac (%)**	**VJ Acc (%)**	**Finetune Ac (%)**
Kinetics	40.3	**49.2**	29.4	**54.7**
UCF101-no oversampling	63.3	46.5	58	46.4
N = no. of permutations	N = 500	N = 500	N = 1000	N = 1000

dataset (without any oversampling), comprising just 200,000 frame tuples and train video jigsaw networks for $N = 500$ and $N = 1000$. The results are shown in Table 5.2. As is shown, even without oversampling, UCF101-based pretraining does not perform as well as Kinetics dataset.

Choice of Number of Permutations We vary the number of permutations N a video jigsaw network has to learn. We start with $N = 100$ and take it up to $N = 1000$. As we increase the number of permutations (see Table 6.2), the network finds it harder to learn the configuration of the patches, but the generalization improves. This experiment is run with Kinetics dataset trained on video jigsaw network.

Choice of Patch Size We vary the patch size sampled from the video frames. Our default patch size was 64×64. As we increase the size (see Table 5.4), the transfer learning improves but only to a certain threshold. Beyond that, the accuracy does not increase. This experiment was performed with our best model with $N = 1000$.

Permutation Generation Strategy We compare the performance of our proposed permutation strategy which enforces spatial coherence (referred to as P_{sp}) between permuted patches — with the proposed approach of [154] (referred to as P_{orig}). We show results for this comparison in Figure 5.4. As the bar chart shows, for various number of permutations,

Table 5.3: As we increase N, the video jigsaw performance decreases but the finetuning accuracy increases

No. of permutations	VJ Acc (%)	Finetuning Ac (%)
100	61.6	44.6
250	44.0	49.0
500	47.6	48.1
1000	29.4	**54.7**

Table 5.4: As we increase patch size, the finetuning accuracy increases upto a certain size, then does not increase further

Patch size	Finetuning Ac (%)
64	54.7
80	**55.4**
100	54.1

our proposed spatial coherency preserving method either outperforms the original random permutation generation strategy or is comparable to it, while being many times faster to generate.

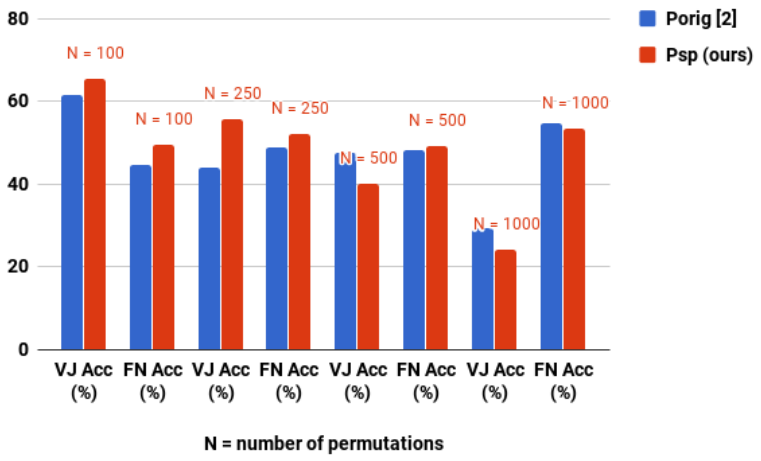

Figure 5.4: Comparison between the permutation strategy proposed by [154] (P_{orig}) and our proposed sampling approach (P_{sp}) on the video jigsaw task (indicated by VJ Acc) and the finetuning task on UCF101 (indicated by FN Acc) for various different number of permutations N. Our approach consistently performs better or comparable to the approach of [154] while saving memory and computational costs. Figure is best viewed in color

Table 5.5: Finetuning results on UCF101 and HMDB51 of our proposed video jigsaw network (pretrained on Kinetics dataset with $N = 1000$ permutations — compared to the state of the art approaches. Note that all these results are computed using *CaffeNet* architecture. Our method gives superior or comparable performance to the state of the art unsupervised learning + finetuning approaches that use RGB frames for training

Pretraining	**UCF101 Acc (%)**	**HMDB51 Acc (%)**
random	40.0	16.3
ImageNet (with labels)	67.7	28.0
Fernando *et al.* [188]	60.3	32.5
Hadsell *et al.* [152]	45.7	16.3
Mobahi *et al.* [129]	45.4	15.9
Wang and Gupta [30]	40.7	15.6
DiscrimNet [196]	50.1	21.0
Misra *et al.* [102]	50.9	19.8
Lee *et al.* [156]	**56.3**	22.1
Vondrick *et al.* [105]	52.1	-
Video Jigsaw Network (ours)	55.4	**27.0**

5.4.3 Finetuning for Action Recognition

Once the video jigsaw network is trained, we use the convolutional layers' weights to initialize a standard *CaffeNet* [10] architecture and use it to finetune on UCF101 and HMDB51 datasets. For UCF101, we sample 25 equidistant frames per video and compute frame-based accuracy as our finetuning evaluation measure. For HMDB51 we sample 1 frame per second from each video and use them for the finetuning experiment. With our best model and parameters (pretrained on Kinetics dataset), results are given in Table 5.5 for test split 1 of both UCF101 and HMDB51 datasets.

Table 5.5 shows our video jigsaw pretraining approach outperforming recent unsupervised pretraining approaches when finetuning on HMDB51 dataset. On UCF101 dataset, our finetuning accuracy is comparable to the state of the art. The method of Fernando *et al.* uses a different input from ours (stacks of frame differences) whereas we use RGB frames to form the jigsaw puzzles. All other approaches operate on RGB video frames or

frame patches hence we can fairly compare with them. The methods of Lee *et al.* [156] and Misra *et al.* [102] are pretrained on UCF101 dataset whereas our best network is trained on Kinetics dataset. This again shows the domain transfer capability of a large scale dataset like Kinetics, compared to UCF101. Our method achieves this without doing any expensive tracking [30] or optical flow based patch or frame mining such as [102, 156]. This means that our approach requires large scale diverse unlabeled video dataset to work. We used 3 frames per video from Kinetics dataset — hence we were only using about 400,000 tuples for our video jigsaw training. We believe that using a larger dataset would lead to better performance, given that our approach is close to the state of the art. Another point to note is that methods which perform well on UCF101 such as Lee *et al.* [156] and Misra *et al.* [102] do not perform that well on HMDB51, whereas our method actually generalizes well, given that it is pretrained on a completely different dataset.

Table 5.6: PASCAL VOC 2007 classification results compared with other methods. Other results taken from [176] and [156]

Method	**Supervision**	**Classification**
ImageNet	1000 class labels	78.2%
Random [186]	none	53.3%
Doersch *et al.* [153]	ImageNet context	55.3%
Jigsaw Puzzle [154]	ImageNet context	67.6%
Counting [176]	ImageNet context	67.7%
Wang and Gupta [30]	100k videos, VOC2012	62.8%
Agrawal *et al.* [178]	egomotion (KITTI, SF)	54.2%
Misra *et al.* [102]	UCF101 videos	54.3%
Lee *et al.* [156]	UCF101 videos	63.8%
Pathak *et al.* [192]	MS COCO + segments	61.0%
Video Jigsaw Network (ours)	Kinetics videos	**63.6%**

5.4.4 Results on PASCAL VOC 2007 Dataset

The PASCAL VOC 2007 dataset consists of 20 object classes with 5011 images in the train set and 4952 images in the test set. Multiple objects can be present in a single image

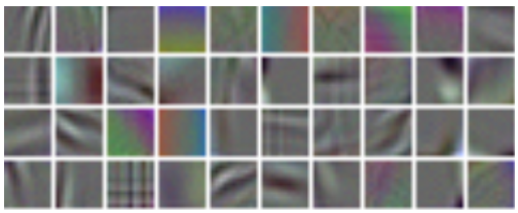

Figure 5.5: Visualization of first 40 learned conv1 filters of our best performing video jigsaw model

and the classification task is to detect whether an object is present in a given image or not. We evaluate our video jigsaw network on this dataset by initializing a *CaffeNet* with our video jigsaw network's trained convolutional layers' weights. The fully-connected layers' weights are randomly sampled from a Gaussian distribution with zero mean and 0.001 standard deviation. Our finetuning scheme follows the one suggested by [197]. Our classification results on the Pascal VOC 2007 test set are shown in Table 5.6.

Our trained network generalizes well not only across datasets but also across tasks. Our video jigsaw network is trained on Kinetics videos and not on object-centric images, yet performs competitively against the state-of-the-art image-based semi-supervised approaches and outperforms most of the video-based semi-supervised methods.

5.4.5 Visualization Experiments

We show first 40 conv1 filter weights of our best video jigsaw model in Figure 5.5 which show oriented edges learned by our model. Note that training this model does not use activity labels. We also perform a qualitative retrieval experiment on the video jigsaw model finetuned on Pascal VOC dataset. Results are shown in Figure 5.6. We note that the retreived images returned by the model match the query image which qualitatively shows that our model trained on unlabeled videos is able to identify objects in still images.

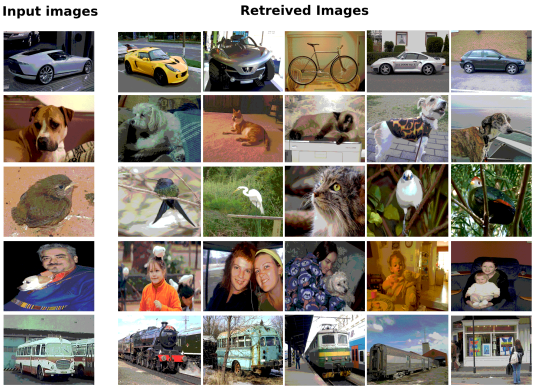

Figure 5.6: Retreival Experiment on PASCAL VOC dataset using our model

5.5 Conclusion

We propose a self-supervised learning task where spatial and temporal contexts are exploited jointly. Our framework is not dependent on heavy preprocessing steps such as object tracking or optical flow based patch mining. We demonstrate via extensive experimental evaluations that our approach performs competitively on video activity recognition, outperforming the state of the art in self-supervised video action recognition on HMDB51 dataset. We also propose a permutation generation strategy which respects spatial coherency and demonstrate that even for shuffling 12 patches, diverse permutations can be generated extremely efficiently via our proposed approach.

CHAPTER 6

EXPLORING THE LIMITS OF SELF-SUPERVISED LEARNING

In this chapter, we present additional results using the self-supervised model described in the previous chapter. In the previous chapter, we used the proposed video jigsaw model and finetuned it on target datasets such as UCF101 [32] and HMDB51 [144]. Unsupervised pretraining + finetuning approaches assume that the target dataset's labels are available. We explore the scenario when none of the target datasets' labels are available; hence the problem becomes that of clustering. Concretely, we address the problem of clustering large-scale real world videos into activities by utilizing our self-supervised model trained on a large unlabeled video dataset. In this chapter, the term 'target dataset' has been used for UCF101 [32] on which we perform clustering.

6.1 Related Work

Current approaches that address clustering of videos into activities use hand-crafted features to learn similarity between videos [198, 199, 200, 201, 202] and test their approaches on small datasets. Our goal is to cluster large-scale video activity datasets such as UCF101 [32] into clusters using deep features. One recent clustering approach that works on larger video datasets is Soomro *et al.* [203] however, it requires the expensive C3D [204] features from a network pretrained on Sports1M [146] dataset as the initial features. Our aim is to explore the limits of 2D CNNs and not require large labeled datasets for extracting initial features.

6.2 Approach

For any unsupervised category discovery (clustering) approach, a similarity metric has to be learned between the data samples where the clustering is performed. This similarity can be learned in many different ways, one of which is to compute features on the data samples, and use a distance metric such as Euclidean distance in order to cluster them. We begin with computing *pool5* features (activations at the pool5 layer of *CaffeNet*) on the video frames; the features are obtained from the trained video jigsaw model described in the previous chapter and use them for clustering using k-means [205] and spectral clustering [206]. Note that there are no labels used in the feature extraction part as well as the clustering part in this experiment.

We further present results where we relax the unsupervised feature extraction condition and use a subset of Kinetics [31] dataset (we call this subset 'Kinetics-101' since we sample 101 categories of Kinetics dataset from its 400 categories) for finetuning our video jigsaw model. This finetuned model is then used to predict labels on UCF101 dataset. We do a careful comparison of the overlap between categories in Kinetics-101 and the categories in the target dataset UCF101 on which we perform the clustering.

6.3 Experiments and Results

In this section, we present our experiments and their outcomes for unsupervised video action discovery in UCF101 dataset.

6.3.1 Visualization

We randomly sample 1 frame per video from the test set of UCF101 for 50 categories, compute pool5 features on all frames and then use Barnes-Hut t-SNE projection [207] to embed the features into 2D and visualize whether the self-supervised video jigsaw model is able to discriminate between action classes (without any finetuning). Figure 6.1 shows

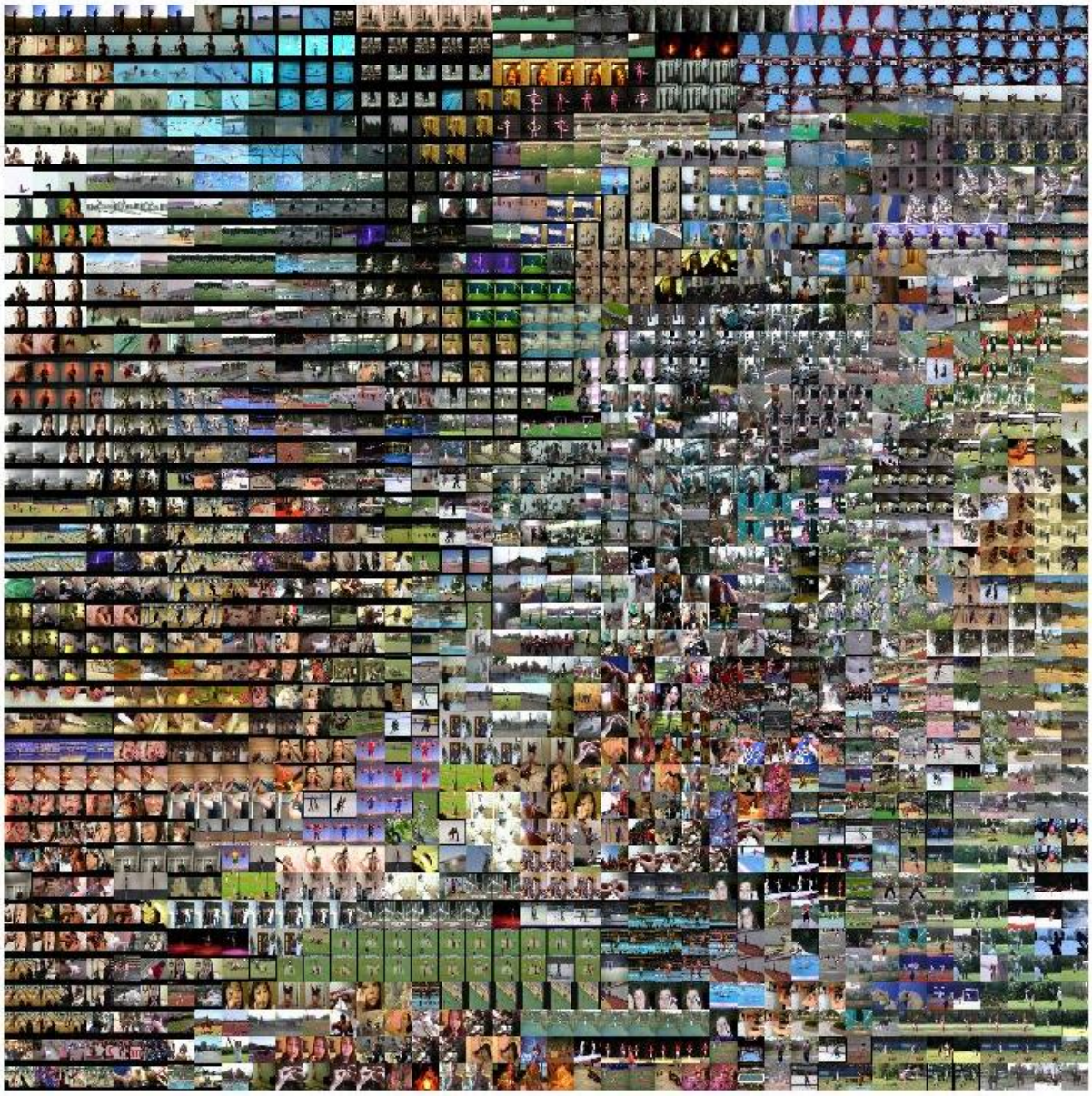

Figure 6.1: Nearest neighbor embedding in which each location is filled by a video frame which is closest to the current frame

the t-SNE embedding [1] where each location contains the nearest neighbor frame to itself. From this zoomed out view of the nearest neighbor embedding, we can quickly observe that local appearance features such as color and shapes are embedded close together. We show zoomed-in views of some of the clusters in Figure 6.2.

[1] https://cs.stanford.edu/people/karpathy/cnnembed/

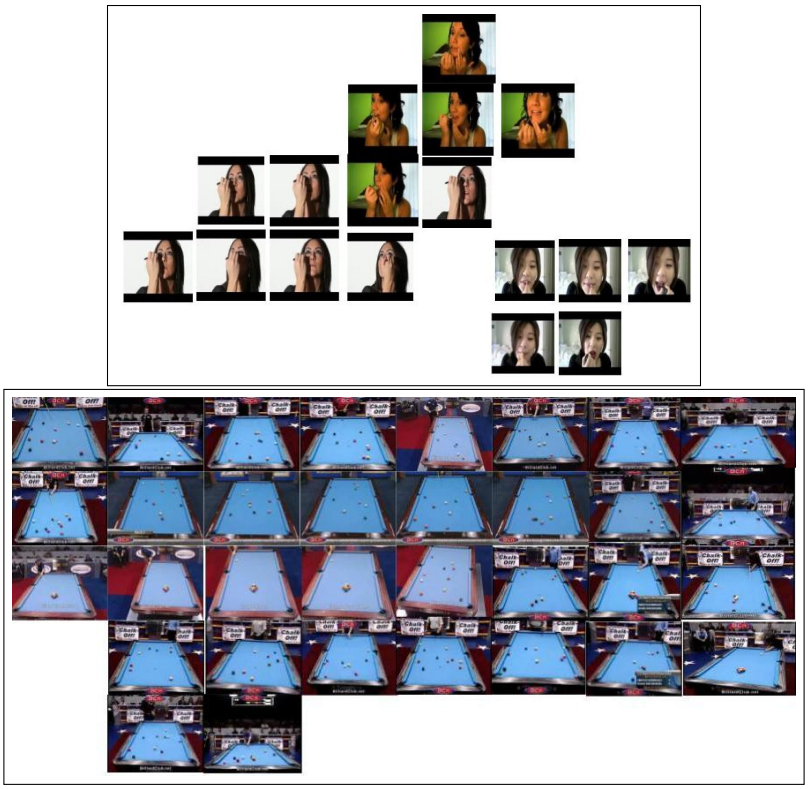

Figure 6.2: Example clusters that are embedded close in t-SNE projection from the self-supervised model's features

6.3.2 Clustering With Self-Supervised Model Features

We extract *pool5* layer's activations on UCF101 dataset by sampling 20 frames from each video and computing features on each frame. To obtain video-level features, we simply take the mean of the frame-level features. Inspired from [202], we also take the number of clusters as equal to the number of ground truth categories and perform clustering using two algorithms: k-means [205] and spectral clustering [206]. We evaluate the clustering results using Normalized Mututal Information (NMI) and clustering accuracy (since we have ground truth available for UCF101 dataset). Our results are shown in Table 6.1. The results indicate that there is a slight improvement to using spectral clustering as opposed to k-means. Since the dataset is not large (9537 training videos and 3783 test videos), it is not expensive to compute the affinity matrix for spectral clustering. Having said that, a crucial assumption in this experiment is the knowledge of number of clusters. The goal

Table 6.1: Comparison between k-means and spectral clustering on *pool5* features on UCF101 videos

Clustering Algorithm	**NMI (train)**	**NMI (test)**	**Accuracy (train %)**	**Accuracy (test %)**
k-means	0.37	0.50	15.93	20.57
Spectral Clustering	**0.39**	**0.52**	**16.38**	**22.42**

of this investigation is to determine how well self-supervised model's features can distinguish between activities in videos without finetuning or any supervision. Once we have a quantifiable measure of how well the self-supervised model performs, next, we relax the completely unsupervised nature of our setup in our experiments. We proceed with conducting a small manual study of the overlap between the categories in Kinetics dataset and the UCF101 dataset.

6.3.3 Kinetics and UCF101 Category Overlap

We conducted a manual study of the categories in Kinetics [31] and UCF101 [32] datasets to determine how similar the activity classes are between the two datasets. This study would help us understand how large the domain shift is when we use Kinetics as the source dataset to cluster activities in UCF101 as the target dataset. Out of 101 categories in UCF101, we found that 87 of those were either exact matches, or highly similar to categories in Kinetics dataset. In other words, 87 out of 400 categories in Kinetics dataset are exactly similar to UCF101 categories. This formed the basis for our next experiment. The goal was to determine how the label overlap affected clustering performance on the target dataset. We therefore chose the 101 labels of Kinetics-101 in the following five ways:

1. **no-overlap (0%)**: Kinetics dataset has 400 categories. We randomly sample 101 categories from the $(400 - 87 = 313)$ non-overlapping categories of Kinetics.

2. **less than 10%**: We randomly sample 101 categories from the 400 categories of Kinetics and determine that there is less than 10% overlap with UCF101 classes.

3. **25%**: We randomly sample 25% of Kinetics-101 categories from the 87 similar cat-

egories and rest from the dissimilar categories.

4. **50%**: Similar to above, we randomly sample 50% of Kinetics-101 categories from the 87 categories and the rest from dissimilar categories.

5. **maximum**: We include all 87 categories in Kinetics-101.

6.3.4 Finetuning On Source + Clustering on Target

We next present results of clustering UCF101 videos into activities by using the self-supervised model finetuned on a subset of Kinetics [31] dataset, which we call 'source dataset'. We sample 101 categories from Kinetics dataset (we call this subset Kinetics-101) and finetune our video jigsaw model on Kinetics-101 frames. We then predict labels on UCF101 frames using this finetuned model. Video-level predictions are computed via majority vote. Since the categories of Kinetics-101 and UCF101 are not aligned, the label predictions are treated as cluster IDs and thus, we evaluate how well the target dataset is partitioned based on these predictions. We perform a thorough analysis on how the clustering performance on UCF101 dataset varies with the label similarity between the Kinetics-101 categories and UCF101 categories. Table 6.2 shows our results. Note that as the category overlap increases between the two datasets, as expected, the clustering performance also increases.

Table 6.2: Varying the % overlap between the labels of source and target dataset and how clustering accuracy changes as a result

% category overlap	**NMI (train)**	**NMI (test)**	**Accuracy (train)**	**Accuracy (test)**
no-overlap (0%)	0.38	0.45	16.04%	18.03%
less than 10%	0.43	0.50	22.40%	23.90%
25%	0.45	0.52	23.30%	26.04%
50%	0.49	0.57	28.10%	31.51%
maximum	**0.56**	**0.62**	**39.20%**	**41.40%**

6.4 Discussion

The experiments conducted in this chapter demonstrate how challenging it is to use self-supervised features for direct activity clustering without finetuning on the target dataset. Even when we use labels of another (source) dataset, action recognition performance on target dataset only increases when there is maximum overlap between the categories of source and target datasets. This is also indicative of the extreme domain shift between the source (Kinetics) and target (UCF101) datasets, even though both datasets have been sampled from YouTube. Explicitly factoring in domain adaptation losses in our approach is a promising future direction. Our visualization experiments indicate that self-supervised models do well in aggregating samples (frames) based on appearance. Higher level semantics need to be either explicitly trained for (for example, finetuning on the dataset of interest) or they need to be learned via domain adaptation.

6.5 Conclusion

In this thesis, we presented a core theme of designing intermediate representations that require minimal supervision, yet are able to generalize to higher level semantic recognition tasks. We design an event concept-based representation that uses abundant textual data on the Web to train concept classifiers that can identify social events. We contribute two event datasets to the community. Further, we demonstrate the generalization capability of event concept representation to recognize sentiment in event images. Next, we propose to use GAN discriminator layer activations as features that can be generalized to recognize activities. We finally propose a self-supervised learning model that is capable of learning representations from large-scale diverse dataset such as Kinetics and generalizes to activity recognition in UCF101 as well as HMDB51 datasets. We also demonstrate the limit of self-supervised models' features and present a study on how these features can be used to cluster activities in videos.

Appendices

APPENDIX A

VIDEO JIGSAW: UNSUPERVISED LEARNING OF SPATIOTEMPORAL CONTEXT FOR VIDEO ACTION RECOGNITION

In this Appendix, we present the original Hamming distance based permutation sampling algorithm proposed by [154] for 12 patches. In Noroozi *et al.* 's impementation, the limit on the number of patches to be permuted was 9. We propose the following algorithm to extend that number to 12. This algorithm is computationally heavy but is able to generate $N = 1000$ permutations for 12 video frame patches.

A.1 Algorithm for Unconstrained Hashes

Algorithm 2 is an alternate implementation of sampling algorithm presented by [154] to deal with spaces larger than $9!$ possibilities. If there are n total patches then instead of loading all $n!$ permutations into the memory which is prohibitive for $n > 9$, we load subsets Λ'' of size $k!$ at a time, where $k < n$ is a loading parameter. The first hash is generated using a random permutation of $\{1, \ldots, n\}$. Each subsequent hash Λ_h $h \in \{2, \ldots, N\}$ is then the permutation from the full set of $n!$ permutation that has the maximum average Hamming distance from all previously generated hashes. This is found by first (a) finding the best hash from each subset Λ'' of $k!$ permutations, which is stored into vector Λ' with its corresponding distance metric into vector D_{max} (lines 5-17), and then (b) finding the best hash from the good ones in Λ' (lines 18-19). Lines 3 and 7-12 describe how each subset Λ'' is constructed. We first define Γ_i, $0 < i < \frac{n!}{(n-k)!k!}$ as sets of all possible combinations of $\{1, \ldots, n\}$ taken k at a time (line 3). Then each subset Λ'' has all $k!$ permutations of a Γ_i (lines 8 and 12). The full permutation (of size n) is obtained by concatenating a particular permutation of the remaining $n-k$ elements with all the $k!$ combinations of Γ_i (lines 9-12).

Algorithm 2: Sampling Permutations without Spatial Coherence

Input: Number of permutations N, total number of patches in all frames n, loading parameter k

Output: Permutation Matrix Λ

function *generatePerm*(N, n, k)
- $\Lambda \leftarrow$ random permutation of $\{1, \ldots, n\}$
- $\Gamma \leftarrow$ all combinations of $\{1, \ldots, n\}$ taken k at a time : $[\Gamma_1, \ldots, \Gamma_{n!/(n-k)!k!}]^\top$
- **for** $h = 2 : N$ **do**
  - $D_{max} \leftarrow \emptyset$
  - $\Lambda' \leftarrow \emptyset$
  - **for** $i = 1 : n!/(n-k)!k!$ **do**
    - $\underset{\rightarrow}{\lambda} \leftarrow$ all permutations of $\Gamma_i : [\underset{\rightarrow}{\lambda}_1, \ldots, \underset{\rightarrow}{\lambda}_{k!}]^\top$
    - $\Upsilon \leftarrow$ all permutations of $\{1, \ldots, n\} - \Gamma_i : [\Upsilon_1, \ldots, \Upsilon_{(n-k)!}]^\top$
    - **for** $j = 1 : (n-k)!$ **do**
      - $\underset{\vdash}{\lambda} \leftarrow [\Upsilon_j, \ldots, \Upsilon_j]^\top \in \mathbb{R}^{k! \times (n-k)!}$
      - $\Lambda'' \leftarrow \begin{bmatrix} \underset{\vdash}{\lambda} & \underset{\rightarrow}{\lambda} \end{bmatrix}^\top$
      - $D \leftarrow Hamming(\Lambda, \Lambda'')$
      - $\overline{D} \leftarrow \frac{1}{h-1} \mathbf{1}^\top D$
      - $D_{max} \leftarrow \begin{bmatrix} D_{max} & \max_k \overline{D}_k \end{bmatrix}$
      - $j \leftarrow \text{argmax}_k \, \overline{D}_k$
      - $\Lambda' \leftarrow \begin{bmatrix} \Lambda' & \Lambda''_j \end{bmatrix}$
    - **end**
  - **end**
  - $j \leftarrow \text{argmax}_k \, D_{max(k)}$
  - $\Lambda \leftarrow \begin{bmatrix} \Lambda & \Lambda'_j \end{bmatrix}$
- **end**
- **return** Λ

end

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, 2015.

[2] G. A. Sigurdsson, O. Russakovsky, and A. Gupta, "What actions are needed for understanding human actions in videos?" *arXiv preprint arXiv:1708.02696*, 2017.

[3] J. Eisenstein, "What to do about bad language on the internet," in *Proceedings of NAACL-HLT*, 2013, pp. 359–369.

[4] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[5] 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," *arXiv preprint arXiv:1609.08675*, 2016.

[6] H. Wechsler, *Neural Networks for Perception: Human and machine perception*. Academic Press, 2014.

[7] K. Ramakrishnan, H. S. Scholte, I. I. Groen, A. W. Smeulders, and S. Ghebreab, "Visual dictionaries as intermediate features in the human brain," *Frontiers in computational neuroscience*, vol. 8, p. 168, 2015.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Visual object detection with deformable part models," *Communications of the ACM*, vol. 56, no. 9, pp. 97–105, 2013.

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 1778–1785.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 951–958.

[12] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 609–623.

[13] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.

[14] D. Parikh and K. Grauman, "Relative attributes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 503–510.

[15] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 365–372.

[16] G. Patterson and J. Hays, "Sun attribute database: discovering, annotating, and recognizing scene attributes," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2751–2758.

[17] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 3337–3344.

[18] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Attribute learning for understanding unstructured social activity," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 530–543.

[19] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133–157, 2007.

[20] G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from flickr groups using stochastic intersection kernel machines," in *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 428–435.

[21] F. Li, J. Carreira, and C. Sminchisescu, "Object recognition as ranking holistic figure-ground hypotheses," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 1712–1719.

[22] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 776–789.

[23] G. Ciocca, C. Cusano, S. Santini, and R. Schettini, "Halfway through the semantic gap: prosemantic features for image retrieval," *Information Sciences*, vol. 181, no. 22, pp. 4943–4958, 2011.

[24] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan, "Zero-shot event detection using multi-modal fusion of weakly supervised concepts," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, 2014, pp. 2665–2672.

[25] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney, "Video event recognition using concept attributes," in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, IEEE, 2013, pp. 339–346.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680.

[27] E. S. Spelke, "Principles of object perception," *Cognitive science*, vol. 14, no. 1, pp. 29–56, 1990.

[28] P. Földiák, "Learning invariance from transformation sequences," *Neural Computation*, vol. 3, no. 2, pp. 194–200, 1991.

[29] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[30] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[31] 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.

[32] 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.

[33] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2556–2563.

[34] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.

[35] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang, "Eventnet: a large scale structured concept library for complex event detection in video," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, ACM, 2015, pp. 471–480.

[36] B. Singh, X. Han, Z. Wu, V. I. Morariu, and L. S. Davis, "Selecting relevant web trained concepts for automated event retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4561–4569.

[37] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann, "Complex event detection via multi-source video attributes," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, IEEE, 2013, pp. 2627–2633.

[38] X. Chen, A. Shrivastava, and A. Gupta, "Neil: extracting visual knowledge from web data," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1409–1416.

[39] S. K. Divvala, A. Farhadi, and C. Guestrin, "Learning everything about anything: webly-supervised visual concept learning," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, 2014, pp. 3270–3277.

[40] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1431–1439.

[41] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM, 2012, pp. 155–164.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[43] B. B. Le Cun, J. S. Denker, D Henderson, R. E. Howard, W Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, Citeseer, 1990.

[44] Q. Li, J. Wu, and Z. Tu, "Harvesting mid-level visual concepts from large-scale internet images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 851–858.

[45] B. Zhou, V. Jagadeesh, and R. Piramuthu, "Conceptlearner: discovering visual concepts from weakly labeled image collections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1492–1500.

[46] C. Sun, C. Gan, and R. Nevatia, "Automatic concept discovery from parallel text and visual corpora," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2596–2604.

[47] L.-J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, IEEE, 2007, pp. 1–8.

[48] V. Jain, A. Singhal, and J. Luo, "Selective hidden random fields: exploiting domain-specific saliency for event classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008, pp. 1–8.

[49] L. Bossard, M. Guillaumin, and L. Van, "Event recognition in photo collections with a stopwatch hmm," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1193–1200.

[50] Y. Xiong, K. Zhu, D. Lin, and X. Tang, "Recognize complex events from static images by fusing deep channels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1600–1609.

[51] E. G. Miller, N. E. Matsakis, and P. A. Viola, "Learning from one example through shared densities on transforms," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, IEEE, vol. 1, 2000, pp. 464–471.

[52] E. Bart and S. Ullman, "Cross-generalization: learning novel classes from a single example by feature replacement," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol. 1, 2005, pp. 672–679.

[53] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 4, pp. 594–611, 2006.

[54] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, vol. 172, 2011, p. 2.

[55] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Advances in neural information processing systems*, 2013, pp. 2526–2534.

[56] G. Koch, "Siamese neural networks for one-shot image recognition," PhD thesis, University of Toronto, 2015.

[57] D. Held, S. Thrun, and S. Savarese, "Deep learning for single-view instance recognition," *arXiv preprint arXiv:1507.08286*, 2015.

[58] A. Wong and A. L. Yuille, "One shot learning via compositions of meaningful patches," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1197–1205.

[59] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang, "Event-driven semantic concept discovery by exploiting weakly tagged internet images," in *Proceedings of International Conference on Multimedia Retrieval*, ACM, 2014, p. 1.

[60] Y. Cui, D. Liu, J. Chen, and S.-F. Chang, "Building a large concept bank for representing events in video," *arXiv preprint arXiv:1403.7591*, 2014.

[61] J. Shao, K. Kang, C. C. Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *Proc. CVPR*, 2015.

[62] X. Yang, T. Zhang, C. Xu, and M. S. Hossain, "Automatic visual concept learning for social event understanding," *Multimedia, IEEE Transactions on*, vol. 17, no. 3, pp. 346–358, 2015.

[63] E. Mezuman and Y. Weiss, "Learning about canonical views from internet image collections," in *Advances in Neural Information Processing Systems*, 2012, pp. 719–727.

[64] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: named entity recognition in targeted twitter stream," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2012, pp. 721–730.

[65] A. Sun and S. S. Bhowmick, "Quantifying tag representativeness of visual content of social images," in *Proceedings of the international conference on Multimedia*, ACM, 2010, pp. 471–480.

[66] D Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, ACM, 2010, pp. 1177–1178.

[67] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.

[68] T. Reuter, S. Papadopoulos, G. Petkos, V. Mezaris, Y. Kompatsiaris, P. Cimiano, C. de Vries, and S. Geva, "Social event detection at mediaeval 2013: challenges, datasets, and evaluation," in *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop Barcelona, Spain, October 18-19, 2013*, 2013.

[69] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[70] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: a library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[71] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: large-scale scene recognition from abbey to zoo," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, IEEE, 2010, pp. 3485–3492.

[72] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: linking text sentiment to public opinion time series.," *ICWSM*, vol. 11, no. 122-129, pp. 1–2, 2010.

[73] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang, "Large-scale visual sentiment ontology and detectors using adjective noun pairs," in *Proceedings of the 21st ACM international conference on Multimedia*, ACM, 2013, pp. 223–232.

[74] U. Ahsan, C. Sun, J. Hays, and I. Essa, "Complex event recognition from images with few training examples," *arXiv preprint arXiv:1701.04769*, 2017.

[75] Y. Hu, F. Wang, and S. Kambhampati, "Listening to the crowd: automated analysis of events via aggregated twitter sentiment," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, AAAI Press, 2013, pp. 2640–2646.

[76] N. A. Diakopoulos and D. A. Shamma, "Characterizing debate performance via aggregated twitter sentiment," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2010, pp. 1195–1198.

[77] M. Hagen, M. Potthast, M. Büchner, and B. Stein, "Twitter sentiment detection via ensemble classification using averaged confidence scores," in *Advances in Information Retrieval*, Springer, 2015, pp. 741–754.

[78] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, pp. 723–762, 2014.

[79] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: tweets as electronic word of mouth," *Journal of the American society for information science and technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[80] M. De Choudhury, M. Gamon, and S. Counts, "Happy, nervous or surprised? classification of human affective states in social media," in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[81] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, "Predicting depression via social media.," in *ICWSM*, 2013.

[82] A. D. Kramer, J. E. Guillory, and J. T. Hancock, "Experimental evidence of massive-scale emotional contagion through social networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 24, pp. 8788–8790, 2014.

[83] S. A. Golder and M. W. Macy, "Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures," *Science*, vol. 333, no. 6051, pp. 1878–1881, 2011.

[84] J. Jia, S. Wu, X. Wang, P. Hu, L. Cai, and J. Tang, "Can we understand van gogh's mood?: learning to infer affects from images in social networks," in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 857–860.

[85] B. Li, S. Feng, W. Xiong, and W. Hu, "Scaring or pleasing: exploit emotional impact of an image," in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 1365–1366.

[86] S. Siersdorfer, E. Minack, F. Deng, and J. Hare, "Analyzing and predicting sentiment of images on the social web," in *Proceedings of the international conference on Multimedia*, ACM, 2010, pp. 715–718.

[87] A. Hanjalic, C. Kofler, and M. Larson, "Intent and its discontents: the user at the wheel of the online video search engine," in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 1239–1248.

[88] J. Yuan, S. Mcdonough, Q. You, and J. Luo, "Sentribute: image sentiment analysis from a mid-level perspective," in *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, ACM, 2013, p. 10.

[89] J. Machajdik and A. Hanbury, "Affective image classification using features inspired by psychology and art theory," in *Proceedings of the international conference on Multimedia*, ACM, 2010, pp. 83–92.

[90] V. Vonikakis and S. Winkler, "Emotion-based sequence of family photos," in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 1371–1372.

[91] Y. Wang, Y. Hu, S. Kambhampati, and B. Li, "Inferring sentiment from web images with joint inference on visual and social cues: a regulated matrix factorization approach," in *Ninth International AAAI Conference on Web and Social Media*, 2015.

[92] T. Chen, D. Borth, T. Darrell, and S.-F. Chang, "Deepsentibank: visual sentiment concept classification with deep convolutional neural networks," *arXiv preprint arXiv:1410.8586*, 2014.

[93] G. Cai and B. Xia, "Convolutional neural networks for multimedia sentiment analysis," in *Natural Language Processing and Chinese Computing*, Springer, 2015, pp. 159–167.

[94] C. Xu, S. Cetintas, K.-C. Lee, and L.-J. Li, "Visual sentiment prediction with deep convolutional neural networks," *arXiv preprint arXiv:1411.5731*, 2014.

[95] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[96] A. Dhall, R. Goecke, and T. Gedeon, "Automatic group happiness intensity analysis," *IEEE Transactions on Affective Computing*, vol. 6, no. 1, pp. 13–26, 2015.

[97] W. Mou, H. Gunes, and I. Patras, "Automatic recognition of emotions and membership in group videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 27–35.

[98] V. Vonikakis, Y. Yazici, V. D. Nguyen, and S. Winkler, "Group happiness assessment using geometric features and dataset balancing," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, 2016, pp. 479–486.

[99] B. Sun, Q. Wei, L. Li, Q. Xu, J. He, and L. Yu, "Lstm for dynamic emotion and group emotion recognition in the wild," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, 2016, pp. 451–457.

[100] J. Li, S. Roy, J. Feng, and T. Sim, "Happiness level prediction with sequential inputs via multiple regressions," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, 2016, pp. 487–493.

[101] J. Wu, Z. Lin, and H. Zha, "Multi-view common space learning for emotion recognition in the wild," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, 2016, pp. 464–471.

[102] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*, Springer, 2016, pp. 527–544.

[103] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," *arXiv preprint arXiv:1611.06646*, 2016.

[104] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[105] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[106] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[107] A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 984–989.

[108] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 1234–1241.

[109] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[110] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conference*, British Machine Vision Association, 2008, pp. 275–1.

[111] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008, pp. 1–8.

[112] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 104–111.

[113] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: action recognition through the motion analysis of tracked features," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 514–521.

[114] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *European Conference on Computer Vision*, Springer, 2012, pp. 425–438.

[115] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[116] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[117] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[118] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[119] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International Workshop on Human Behavior Understanding*, Springer, 2011, pp. 29–39.

[120] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[121] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[122] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, X. Xue, and J. Wang, "Fusing multi-stream deep networks for video classification," *arXiv preprint arXiv:1509.06086*, 2015.

[123] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[124] Y. Zhou and T. L. Berg, "Temporal perception and prediction in ego-centric video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4498–4506.

[125] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating the future by watching unlabeled video," *arXiv preprint arXiv:1504.08023*, 2015.

[126] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun, "Unsupervised learning of spatiotemporally coherent metrics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4086–4093.

[127] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms," *CoRR, abs/1502.04681*, vol. 2, 2015.

[128] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[129] H. Mobahi, R. Collobert, and J. Weston, "Deep learning from temporal coherence in video," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 2009, pp. 737–744.

[130] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European conference on computer vision*, Springer, 2010, pp. 140–153.

[131] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 436–450, 2012.

[132] D. Jayaraman and K. Grauman, "Slow and steady feature analysis: higher order temporal coherence in video," *arXiv preprint arXiv:1506.04714*, 2015.

[133] X. Wang, A. Farhadi, and A. Gupta, "Actions$\tilde{}$transformations," *arXiv preprint arXiv:1512.00795*, 2015.

[134] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[135] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[136] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: interpretable representation learning by information maximizing generative adversarial nets," *arXiv preprint arXiv:1606.03657*, 2016.

[137] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *arXiv preprint arXiv:1606.03498*, 2016.

[138] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[139] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[140] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[141] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[142] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[143] I. Goodfellow, "Nips 2016 tutorial: generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

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

[145] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 248–255.

[146] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[147] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[148] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[149] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European Conference on Computer Vision*, Springer, 2016, pp. 646–661.

[150] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[151] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.."

[152] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, IEEE, vol. 2, 2006, pp. 1735–1742.

[153] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[154] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*, Springer, 2016, pp. 69–84.

[155] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," *arXiv preprint arXiv:1708.02901*, 2017.

[156] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, "Unsupervised representation learning by sorting sequences," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 667–676.

[157] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision*, Springer, 2016, pp. 649–666.

[158] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*, Springer, 2016, pp. 577–593.

[159] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2051–2060.

[160] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[161] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[162] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 1096–1103.

[163] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 693–700.

[164] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801–808.

[165] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[166] G. E. Hinton and T. J. Sejnowski, "Learning and releaming in boltzmann machines," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, no. 282-317, p. 2, 1986.

[167] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, Tech. Rep., 1986.

[168] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, IEEE, vol. 1, 2005, pp. 370–377.

[169] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, IEEE, vol. 2, 2006, pp. 1605–1614.

[170] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in neural information processing systems*, 2013, pp. 494–502.

[171] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Computer Vision–ECCV 2012*, Springer, 2012, pp. 73–86.

[172] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, pp. 3400–3407.

[173] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural net-

works," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1734–1747, 2016.

[174] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer, "Cliquecnn: deep unsupervised exemplar learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3846–3854.

[175] M. A. Bautista, A. Sanakoyeu, and B. Ommer, "Deep unsupervised similarity learning using partially ordered sets," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2017.

[176] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5898–5906.

[177] D. Jayaraman and K. Grauman, "Learning image representations tied to ego-motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1413–1421.

[178] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, IEEE, 2015, pp. 37–45.

[179] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *European Conference on Computer Vision*, Springer, 2016, pp. 801–816.

[180] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2405–2413.

[181] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: experiential learning of intuitive physics," in *Advances in Neural Information Processing Systems*, 2016, pp. 5074–5082.

[182] L. Pinto, J. Davidson, and A. Gupta, "Supervision via competition: robot adversaries for learning tasks," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1601–1608.

[183] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The curious robot: learning visual representations via physical interactions," in *European Conference on Computer Vision*, Springer, 2016, pp. 3–18.

[184] L. Pinto and A. Gupta, "Learning to push by grasping: using multiple tasks for effective learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 2161–2168.

[185] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[186] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[187] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1058–1067.

[188] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 5729–5738.

[189] T. Milbich, M. Bautista, E. Sutter, and B. Ommer, "Unsupervised video understanding by reconciliation of posture similarities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4394–4404.

[190] B. Brattoli, U. Buchler, A.-S. Wahl, M. E. Schwab, and B. Ommer, "Lstm self-supervision for detailed behavior analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6466–6475.

[191] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 766–774.

[192] D. Pathak, R. Girshick, P. Dollar, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2701–2710.

[193] Z. Ren and Y. J. Lee, "Cross-domain self-supervised multi-task feature learning using synthetic imagery," *arXiv preprint arXiv:1711.09082*, 2017.

[194] O. Sumer, T. Dencker, and B. Ommer, "Self-supervised learning of pose embeddings from spatiotemporal relations in videos," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 4308–4317.

[195] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 41–48.

[196] U. Ahsan, C. Sun, and I. Essa, "Discrimnet: semi-supervised action recognition from videos using generative adversarial networks," *arXiv preprint arXiv:1801.07230*, 2018.

[197] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, "Data-dependent initializations of convolutional neural networks," *arXiv preprint arXiv:1511.06856*, 2015.

[198] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei, "Spatial-temporal correlatons for unsupervised action classification," in *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, IEEE, 2008, pp. 1–8.

[199] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008.

[200] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 102–114, 2017.

[201] S. Jones and L. Shao, "A multigraph representation for improved unsupervised/semi-supervised learning of human actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 820–826.

[202] ——, "Unsupervised spectral dual assignment clustering of human actions in context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 604–611.

[203] K. Soomro and M. Shah, "Unsupervised action discovery and localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 696–705.

[204] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[205] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[206] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[207] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.