UNDERSTANDING ACTIONS IN INSTRUCTIONAL VIDEOS

EĞİTİCİ VİDEOLARDAKİ EYLEMLERİ ANLAMAK

ÖZGE YALÇINKAYA ŞİMŞEK

PROF. DR. PINAR DUYGULU ŞAHİN
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Doctor of Philosophy
in Computer Engineering

May 2023

ABSTRACT

UNDERSTANDING ACTIONS IN INSTRUCTIONAL VIDEOS

Özge Yalçınkaya Şimşek
Doctor of Philosophy, Computer Engineering
Supervisor: Prof. Dr. Pınar Duygulu Şahin
May 2023, 103 pages

Despite the promising improvements in human activity recognition for fundamental actions, understanding actions in instructional web videos is a challenging research problem. Instructional web videos contain various demonstrations for daily human tasks such as cooking, repairing, or *how to do something*. These tasks involve multiple fine-grained and complex actions in different orders or appearances according to the people demonstrating them. In addition to the challenge of finding a robust action localization model for fine-grained actions, a vast amount of labeled data is needed.

Due to the difficulties in labeling such data, studies propose self-supervised video representation learning models that leverage the narrations in the videos. By utilizing the automatically generated transcriptions obtained from the narrators, joint embedding spaces are trained to learn video-text similarities. Hence, pre-defined instructional action steps can be localized, leveraging the video-text similarity information.

However, learning such models presents a drawback where background video clips containing non-action scenes can get high scores for specific action steps due to the misleading transcriptions paired with them in training. Therefore, action localization results

can be impacted. Detecting harmful backgrounds and distinguishing them from action clips is essential.

In this thesis, we investigate improving action localization results on instructional videos by defining actions and backgrounds with a novel representation that forces their discrimination in the joint feature space. We show that using baseline video-text model similarity score cues to describe each video clip reinforces the discrimination of action clips and backgrounds. The idea is based on the hypothesis that background video clips tend to obtain uniformly distributed low similarity scores for all action step labels, whereas action clips get high scores for specific action steps.

We jointly train a binary model that encodes the *actionness* of each video clip with this discriminative representation and visual features. Here the actionness score defines the probability for a given video clip to contain an action. Then, we use the actionness scores of each video clip to post-process action localization and action segmentation scores of baseline video-text models. We present results on CrossTask and COIN datasets.

We show that a small labeled set is sufficient for action/background discrimination learning. However, the quality of the data affects more. Thus, we investigate the effect of augmenting training data with action images collected from the web and utilize an image-text model. We show promising results for future directions along with challenges due to the bottleneck of the model and dataset.

Keywords: instructional videos, action localization, action segmentation, actionness, video representation, action images

ÖZET

EĞİTİCİ VİDEOLARDAKİ EYLEMLERİ ANLAMAK

Özge Yalçınkaya Şimşek
Doktora, Bilgisayar Mühendisliği
Danışman: Prof. Dr. Pınar Duygulu Şahin
Mayıs 2023, 103 sayfa

Temel eylemler için insan etkinliği tanımadaki umut verici gelişmelere rağmen, eğitici internet videolarındaki eylemleri anlamak zorlu bir araştırma problemidir. Eğitici internet videoları, yemek pişirme, tamir etme veya *bir şey nasıl yapılır* gibi günlük insan görevleri için çeşitli gösterimler içerir. Bu görevler, onları yapan insanlara göre değişmek üzere farklı sırada veya görünümde çok sayıda ince taneli ve karmaşık eylem içermektedir. İnce taneli eylemler için sağlam bir eylem yerelleştirme modeli bulma zorluğuna ek olarak, büyük miktarda etiketlenmiş veriye ihtiyaç vardır.

Bu tür verileri etiketlemenin zorlukları nedeniyle, araştırmalar, videolardaki anlatımlardan yararlanan ve kendi kendini denetleyen video temsili öğrenme modelleri önermiştir. Anlatıcılardan elde edilen otomatik olarak oluşturulmuş altyazılardan yararlanılarak, video-metin benzerliklerini öğrenmek için ortak özellik uzayları eğitilir. Bu sayede, video-metin benzerlik bilgisinden yararlanılarak önceden tanımlanmış eğitimsel eylem adımları yerelleştirilebilir.

Bununla birlikte, bu tür modellerin öğrenilmesi, eylem dışı sahneler içeren arka plan video kliplerinin, eğitim sırasında eşleştikleri yanıltıcı altyazılar nedeniyle eylem adımları için yüksek puanlar alabildiği durumlarda bir dezavantaj sunar. Bu nedenle, eylem yerelleştirme

sonuçları etkilenebilir. Zararlı arka planları tespit etmek ve bunları eylem kliplerinden ayırmak önemli olmaktadır.

Bu tezde, eylemlerin ve arka planların ortak özellik uzayındaki ayrımlarını zorlayan yeni bir temsil tanımlayarak, eğitici videolarda eylem yerelleştirme sonuçlarının iyileştirilmesini araştırıyoruz. Her bir video klibi tanımlamak için temel video-metin modeli benzerlik puanı ipuçlarını kullanmanın, eylem klipleri ve arka planların ayrımını güçlendirdiğini gösteriyoruz. Bu fikir, arka plan video kliplerinin tüm eylem adımı etiketleri için düşük benzerlik puanları alma eğiliminde olduğu ve karşıt olarak eylem kliplerinin belirli eylem adımları için yüksek puanlar aldığı hipotezine dayanmaktadır.

Her video klibin *eylemliliğini* bu ayırt edici temsil ve görsel özelliklerle birlikte kodlayan bir ikili modeli ortaklaşa eğitiyoruz. Burada eylemlilik puanı, belirli bir video klibin bir eylem içerme olasılığını tanımlar. Ardından, temel video-metin modellerinin eylem yerelleştirme ve eylem segmentasyonu puanlarını sonradan işlemek için her bir video klibin eylemlilik puanlarını kullanıyoruz. Sonuçları CrossTask ve COIN veri kümeleri üzerinde sunuyoruz.

Eylem/arka plan ayrımcılığının öğrenilmesi için küçük etiketli bir kümenin yeterli olduğunu göstermenin ardından veri kalitesinin sonuçları daha fazla etkileyebileceğini tartışıyoruz. Ek bir çözüm olarak, internetten toplanan eylem görüntüleriyle eğitim setini çeşitlendirmenin etkisini araştırıyoruz ve bir görüntü-metin modeli kullanıyoruz. Modelin ve veri kümesinin darboğazından kaynaklanan zorluklarla birlikte gelecekteki yönelimler için umut verici sonuçlar gösteriyoruz.

Keywords: eğitici videolar, eylem yerelleştirme, eylem segmentasyonu, eylemlilik, video temsili, eylem görüntüleri

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Prof. Dr. Pinar Duygulu Sahin, for her constant guidance, support, and encouragement throughout this long journey filled with many ups and downs.

I want to express my gratitude to Asst. Prof. Dr. Olga Russakovsky for her great contribution to this thesis. Her support and teaching during a pandemic mean a lot, and I've learned so much from her about doing research. Furthermore, I want to thank Princeton Visual AI Lab members for sharing their knowledge which contributed to my research perspective.

I would like to thank my thesis monitoring committee members Asst. Prof. Dr. Ramazan Gokberk Cinbis and Assoc. Prof. Dr. Erkut Erdem for their valuable comments, guidance, and ideas, which contributed to the course of this thesis.

Moreover, I thank my thesis defense jury members Assoc. Prof. Dr. Sinan Kalkan, Assoc. Prof. Dr. Lale Ozkahya and Asst. Prof. Dr. Engin Demir for accepting to review and evaluate this thesis.

Finally, I want to thank my family (Oya, Abdurrahman, Anil) and friends (Nihan, Buket, Yagmur), who always support me when needed. The biggest thanks go to my husband, Mert, who got through a lot by being just by my side and helping me all the way.

CONTENTS

Page

ABSTRACT i

ÖZET iii

ACKNOWLEDGEMENTS v

CONTENTS vi

TABLES ix

FIGURES xi

ABBREVIATIONS xiv

1. INTRODUCTION 1
   - 1.1. Scope Of The Thesis 3
   - 1.2. Contributions 5
   - 1.3. Organization 6
2. RELATED WORK 7
   - 2.1. Self-supervised Video Representation Learning 7
   - 2.2. Action Localization on Instructional Videos 9
   - 2.3. Actionness Learning 11
   - 2.4. Image to Video Adaptation 13
3. ACTION LOCALIZATION AND ACTION/BACKGROUND DISCRIMINATION ON INSTRUCTIONAL VIDEOS 14
   - 3.1. Action Localization on CrossTask with MIL-NCE 15
     - 3.1.1. CrossTask 15
     - 3.1.2. MIL-NCE 15
     - 3.1.3. Action Localization on CrossTask 16
   - 3.2. Action/Background Discrimination with Similarity Cues 19
   - 3.3. Discussion 20
4. LEARNING ACTIONNESS FROM ACTION/BACKGROUND DISCRIMINATION 22
   - 4.1. Method 24
     - 4.1.1. Learning Actionness 24

4.1.2. Post-processing with Actionness 26
4.2. Datasets 28
4.3. Implementation Details 29
4.4. Experiments 29
4.4.1. Action Step Localization 29
4.4.2. Action Segmentation 31
4.4.3. Ablation Studies 32
4.4.3.1. Visual features: 32
4.4.3.2. Auxiliary loss: 32
4.4.3.3. Neighbors in auxiliary representation: 33
4.4.3.4. Using 4-dimensional auxiliary representation: 34
4.4.3.5. Amount of training samples: 34
4.5. Discussion 35
5. IMPROVING ACTION/BACKGROUND DISCRIMINATION WITH IMAGES 38
5.1. Comprehensive Analysis and Discussion of the CrossTask-Cooking Dataset 40
5.1.1. Challenging Action Step Labels 41
5.1.2. Challenging Annotations 46
5.1.3. The Number of Positive and Background Data 49
5.2. Datasets 50
5.2.1. Google Images Collected with Modified Labels 50
5.2.2. RecipeQA 51
5.3. Implementation Details 53
5.4. Experiments 53
5.4.1. CLIP Baseline 53
5.4.2. Hand-selected Subset of Google Images 54
5.4.2.1. Quality of the Examples: 55
5.4.2.2. Results: 57
5.4.3. Auto-selected Subsets of RecipeQA and Google Images 61
5.5. Discussion 62
6. CONCLUSION 64

6.1. Limitations and Future Directions 65

TABLES

Page

Table 3.1 The baseline and upper bound step recall results for each task when background clips get the lowest scores. The evaluation set is val-test. . . 18

Table 4.1 # of videos included and usage of different splits. 29

Table 4.2 **Action Step Localization.** Post-processing with actionness scores of the discrimination model d increases the step recall (SR) on CrossTask. The method in [1] improves the same baseline [2]. * depicts the reproduced baseline. 30

Table 4.3 Task-based CrossTask action step localization results (SR) and the ratio of corrections done from false positives on the backgrounds to true positives. 31

Table 4.4 **Action Segmentation.** Frame-wise accuracy (FA) results on COIN after applying post-processing with actionness scores of d. * depicts the reproduced baseline. 32

Table 4.5 Various visual features are employed to train d with CrossTask val-test set. The average Precision (AP) metric measures the discrimination between actions and backgrounds, while the Step Recall (SR) is evaluated following post-processing with the trained d. 32

Table 4.6 Different configurations of the actionness model d and their impact on the performance metrics of average precision (AP) for action/background discrimination and step recall (SR). One such configuration involves employing random numbers generated from a uniform distribution as the actionness score. 33

Table 5.1 Average precision (AP) of action/background discrimination and average step recall (SR) results for CrossTask-Cooking dataset with CLIP image-based features. 54

Table 5.2 Average precision (AP) of action/background discrimination and average step recall (SR) results when val-train is augmented with different hand-selected Google image sets. 58
Table 5.3 Average step recall results for 4 action step classes when image augmentation with the hand-selected “Best 4” is applied. 59
Table 5.4 Average step recall results of some improved action step classes when image augmentation with the hand-selected “Steam milk” images is applied. 60
Table 5.5 Results for additional training set variants. 61
Table 5.6 Average precision (AP) of action/background discrimination and average step recall (SR) results when val-train is augmented with auto-selected image sets collected according to text-to-image similarities. 62

FIGURES

Page

Figure 1.1 An instructional web video that contains multiple fine-grained action steps to accomplish the task "how to make a cake"..... 3

Figure 3.1 The baseline model trained with the MIL-NCE loss. This figure is taken from [2]..... 17

Figure 3.2 Components calculated from the video clip $x \in X$ to action step label $k \in K_D$ assignment probability cues. Final vector is formed as $\mathbf{r_x}$..... 20

Figure 3.3 Linear SVM weights assigned to the components of $\mathbf{r_x}$..... 21

Figure 4.1 Overview of our method. We learn a discrimination model d to derive an actionness score $d(x)$, indicating the probability of an action being present in the given input x. Leveraging the output of the baseline video-to-action outputs $f_k(x)$ for each video clip x and action step label k, we integrate the actionness score with the baseline scores to obtain the post-processed scores $f_k^{'}(x)$. Specifically, the proposed approach corrects false positive predictions on background clips, such as the "pour water" action step in the illustrative example, and elevates the scores of action clips to identify the associated action step label accurately. 23

Figure 4.2 Proposed auxiliary representation denoted as $\mathbf{a_x}$ for a video clip x. The representation is constructed using the probabilities $f_k(x)$ of assigning action step labels to a video clip x. The temporal window for selecting neighboring probabilities is set to $t = 3$. 24

Figure 4.3 The proposed discrimination model d to obtain the actionness of a video clip x. It is jointly trained with auxiliary $\mathbf{a_x}$ and visual $\mathbf{v_x}$ features. The actionness $d(x)$ is the probability of being an action for the visual representation $\mathbf{v_x}$..... 25

Figure 4.4 We present the predicted video clips for different action step labels before (baseline) and after post-processing with d. Our actionness score converts false positive predictions into true positives. 31
Figure 4.5 Examples to where post-processing with d is better than d_v trained with only visual loss L_v. 34
Figure 4.6 The effect of varying CrossTask val-train sets on post-processing. The baseline set contains 12 videos per task; its step recall is 44.20. .. 35
Figure 4.7 False predictions after post-processing with d. Coarse annotation of the dataset could be effective adversely as training of the d depends on the labels. 36
Figure 5.1 Different styles of executions for the “add strawberries to cake” action. 38
Figure 5.2 Example frames for “add coffee”. 42
Figure 5.3 Example frames for “add ice”. 42
Figure 5.4 Example frames for “add meat”. 43
Figure 5.5 Example frames for “add taco”. 43
Figure 5.6 Example frames for “add tortilla”. 43
Figure 5.7 Example frames for “add whipped cream”. 44
Figure 5.8 Example frames for “close lid” and “open lid”. 44
Figure 5.9 Example frames for “mix ingredients”. 45
Figure 5.10 Example frames for “pour egg”. 45
Figure 5.11 Example frames for “put dough into form”. 46
Figure 5.12 Example frames for “pour water”. 46
Figure 5.13 Example video clips for specific actions where the narrators are annotated with action step labels. 47
Figure 5.14 Examples of faulty annotations. 48
Figure 5.15 Examples of discrepancies between the action in the video and the annotation. 49
Figure 5.16 The number of samples for each action step in the CrossTask-Cooking val-train. 49
Figure 5.17 The total number of Google images collected for each action step label. 51

Figure 5.18 Example of top Google images for “squeeze lemon”..... 52
Figure 5.19 Example instructional step images for “3 minutes vegan cake” recipe in the RecipeQA [3]. 52
Figure 5.20 The number of total samples in the val-train set before and after augmenting the number of clips with the hand-selected subset of Google Images..... 55
Figure 5.21 Original video frames and difficult examples selected from Google Images for action steps. 56
Figure 5.22 Representative Google images selected for action steps..... 57
Figure 5.23 Comparison of step recall results for “Baseline (d_v [4])” trained with val-train and “All” trained with val-train + hand-selected Google images..... 58
Figure 5.24 Example video frames for specific action step classes where hand-selected “steam milk” images contribute to their discrimination. 60
Figure 5.25 The number of clips chosen by using hand-selected Google images as seeds..... 61
Figure 5.26 Example images selected with text-to-image similarity values of CLIP. 62

ABBREVIATIONS

AP	:	**A**verage **P**recision
API	:	**A**pplication **P**rogramming **I**nterface
ASR	:	**A**utomatic **S**peech **R**ecognition
CLIP	:	**C**ontrastive **L**anguage-**I**mage **P**re-Trainin
CNNs	:	**C**onvolutional **N**eural **N**etworks
COOT	:	**Coo**perative hierarchical **T**ransformer
CT	:	**C**ross**T**ask
FP	:	**F**alse **P**ositive
fps	:	**f**rame **p**er **s**econd
I3D	:	Two-Stream **I**nflated **3D**imensional ConvNets
LinearSVC	:	**Linear** **S**upport **V**ector **C**lassifier
MIL-NCE	:	**M**ultiple **I**nstance **L**earning - **N**oise **C**ontrastive **E**stimation
S3D	:	**S**eparable **3D**imensional CNN
SR	:	**S**tep **R**ecall
SVM	:	**S**upport **V**ector **M**achine
TP	:	**T**rue **P**ositive
video-to-action probability	:	**video** clip to **action** step label assignment **probabilities**
ViT	:	**Vi**sion **T**ransformer

1. INTRODUCTION

The development of visual recognition systems has undergone significant advancements due to the introduction of improved machine learning models. In particular, object, scene, and facial recognition have become less challenging with the enhancements made on Deep Neural Networks. On the other hand, recognition of human activities, which require an understanding of multiple scenes in a temporal order, still poses an open research area, despite the important achievements. Such task is necessary to enable machines to learn human activities for various purposes, such as video retrieval, household robot teaching, anomaly detection, assistance systems for industrial applications, and instruction-specific warning systems, and more.

Video-based training is an effective approach for modeling human actions. In order to learn a representation for an action, researchers exposed machine learning systems to multiple example activities captured through video frames or scenes. Initial investigations involved learning six hand-crafted actions from short videos, each containing only one action [5]. Then, more realistic actions were learned by exploring movie scenes [6]. With the vast number of videos available online, researchers have proposed utilizing YouTube videos to create large-scale activity recognition datasets, which are widely regarded as benchmarks in the field [7–10]. The emergence of Convolutional Neural Networks (CNNs) for object recognition has paved the way for their application in human activity recognition [11]. This has led to the development of deep architectures that leverage temporal and spatial information to understand videos [10, 12–17]. In recent years, there has been a shift towards transformer-based deep models, which has also influenced activity recognition models [18, 19].

While the aforementioned deep architectures demonstrate significant success in understanding fundamental actions, learning daily life human activities such as cooking, crafting, or repairing remained in development. These activities involve multiple fine-grained action steps in sequential order, are often complex, and have wide intra-class variety. They

are demonstrated in a hierarchical structure, starting with a phase like grasping a cup and progressing to subsequent steps such as pouring water. Each individual action entails multiple interconnected steps that can be elucidated by illustrating transformations. For instance, the process of preparing tomatoes involves sequentially holding them, grasping a knife, cutting the tomatoes into pieces, and adding them to a bowl. Consequently, it is necessary for researchers to move beyond the conventional approach of assigning a single action label to a video and training models accordingly. Instead, they require datasets with fine-grained labels and methodologies capable of localizing multiple actions within a video. This novel challenge in the field of human activity recognition is commonly referred to as *understanding instructional videos*, which encompasses the demonstration of multiple instructions for daily human activities in video datasets.

Many studies investigate learning and encoding hierarchical structures and transformations that happen in instructional videos to understand daily-human activities [20–26]. Furthermore, numerous studies have employed labeled and hand-crafted videos of cooking or daily activities to detect and localize fine-grained pre-defined actions [27–30]. However, given the varying environments, distinct camera viewpoints, differently used objects, and different demonstrations of the same action, researchers have sought more diverse and realistic data to improve learning. Instructional YouTube videos have thus gained attention, as they often demonstrate *how to do something* in great detail [31, 32]. Nonetheless, labeling long-duration videos with fine-grained actions poses a bottleneck. For instance, the instructional YouTube videos in CrossTask dataset have an average duration of 4 minutes and 57 seconds, and 72% of the frames are backgrounds while there are only 11 action steps at most [33][1]. Therefore, in such a challenging setting, it is difficult both for humans and machine learning models to label or recognize corresponding actions appropriately.

As a solution to labeling problem, automatically generated transcripts from the narrations are utilized [34]. An example video sequence is given in Figure 1.1. The idea is to use narrations as the descriptions of related video clip chunks to provide weak supervision. Then, self-supervised video representation models have been developed to learn a joint

[1] https://arxiv.org/pdf/1903.08225.pdf

embedding space between video clips and transcriptions [2, 35]. These models aim to force the encodings of a video clip and its text to be close in the embedding space while making the others far apart. As a result, video-to-text or text-to-video retrieval tasks are improved, along with action localization and segmentation.

Figure 1.1 An instructional web video that contains multiple fine-grained action steps to accomplish the task "how to make a cake".

The utilization of transcriptions presents certain limitations, including their sensitivity to background video clips unrelated to the instructional task but featuring the demonstrator or irrelevant scenes. This sensitivity primarily results from the misalignment between the video clip and its corresponding description, which is inevitable in YouTube videos, as individuals may describe their actions before or after demonstrating them. As a solution, in [2], researchers propose to leverage surrounding transcriptions to find the best matching video-text pairs before training the model. However, there remain instances of false positive predictions owing to the high similarity scores between background scenes and related texts. Therefore, detecting backgrounds and decreasing their misleading scores is essential for improving the localization of the actions. One way is to learn the discrimination between background and action video clips by using the representations from video-text models. In this thesis, we aim to address the issue of mitigating the adverse effects of background video clips in downstream tasks by proposing the discrimination of such clips from action clips with the usage of video-text similarity cues along with the baseline representations.

1.1. Scope Of The Thesis

This thesis focuses on improving two downstream tasks, namely action localization and segmentation on instructional web videos, through the discrimination of action and

background video clips. These tasks necessitate identifying and localizing a predefined set of action steps in a given video. In this study, the CrossTask [33] and COIN [36] datasets are used to investigate the robustness of current video-text models against background video clips that may mislead the action assignments, leading to false positive predictions. To address this issue, we explore using the MIL-NCE [2] and COOT [37] models to represent and distinguish background and action clips. We propose a method that relies on baseline video to action step label similarity scores as an indicator of discrimination. Our hypothesis is that while background clips yield uncertain scores for all steps, action clips produce more confident scores for specific action steps.

Following the demonstration of the usefulness of video to action step label similarities in detecting backgrounds, our investigation aims to leverage this information to enhance downstream tasks. Prior research by Zhukov et al. [1] proposes using an **actionness** score for each video clip to indicate its probability of containing an action. This score is employed to adjust the baseline similarity scores and modify the action assignments correspondingly. Inspired by this approach, we introduce a novel actionness learning model, which incorporates our action/background discrimination method, and demonstrate its impact on improving the performance of two distinct tasks. We present this work in the below study:

- O. Yalcinkaya Simsek, O. Russakovsky and P. Duygulu, "Learning actionness from action/background discrimination", Signal, Image and Video Processing (SIVP), pp.1-8, 2022.

Despite the advances shown in our work, we further analyze the outcomes and the dataset employed to train our model, CrossTask. Our findings indicate that the coarse labeling of actions and backgrounds could compromise the efficiency of our model training. As such, we present a detailed exploration of the labeling methodology of the CrossTask dataset, which has served as a standard benchmark for action localization tasks over time. Drawing on the insights of prior research [38], we suggest including images to improve the quality

of training data. Specifically, we aim to enhance the action/background discrimination model by utilizing a more dependable training dataset. To this end, we change our baseline model with the state-of-the-art image classification model CLIP [39] and use video frames instead of clips. We curate action images from Google using the predefined action step labels of CrossTask, and we also incorporate instructional images from the RecipeQA dataset [3]. Our results suggest that collecting and integrating action images for better discrimination is challenging but provides an opportunity for future research. Nevertheless, preliminary investigations indicate that augmenting the training data with confident examples can improve action localization.

1.2. Contributions

The main contributions of this thesis are summarized as follows:

- We explore the adverse effect of background video clips on action localization task for instructional web videos.
- We propose defining action and background video clips by the cues computed from their similarity scores to the action step labels.
- We propose a new model to learn **actionness** of video clips for enhancing downstream tasks. Our model performs better than the state-of-the-art study for improvement over the same baseline.
- We propose a novel post-processing strategy for the action segmentation task.
- We present a detailed analysis of the CrossTask's annotations.
- We change our representations and work in the image-based domain to augment the training set with action images in order to enhance action/background discrimination. We present promising results for future works along with the challenges.

1.3. Organization

The organization of the thesis is as follows:

- **Chapter 2** provides a detailed literature review on self-supervised video representation learning, action localization on instructional videos, actionness learning, and image-to-video learning.
- **Chapter 3** details the primary baseline model used in this work, MIL-NCE [2], and the downstream task, action localization. After that, the utilization of video to action step similarity scores for learning action/background discrimination is explored.
- **Chapter 4** introduces our previous work [4], which covers learning actionness from an action/background discrimination model and its usage for post-processing the baseline scores.
- **Chapter 5** demonstrates the detailed analysis of the CrossTask [33] dataset's annotations and proposes using action images to augment the training set for an improved action/background discrimination.
- **Chapter 6** states the thesis summary and possible future directions.

2. RELATED WORK

This study uses self-supervised video representation models that focus on learning video-text embedding spaces. We leverage the encodings taken from these models to accomplish action localization task on instructional videos. Then, in order to enhance the performance, we explore learning the actionness of video clips by discriminating actions and backgrounds. Finally, we investigate improving the discrimination of clips with action images. In this section, we give a detailed literature review on the mentioned research topics.

2.1. Self-supervised Video Representation Learning

In order to understand complex human actions, the studies for video representation learning explore the utilization of large-scale web videos. Due to the inevitable challenge of labeling such videos, self-supervised architectures are developed where the models learn a joint embedding space between video clips and texts by leveraging the narrations of the videos. The achievements are shown by evaluating these models' video and text encodings on multiple downstream tasks such as action recognition, action localization, action segmentation, video captioning, and text-to-video retrieval.

Miech et al. [34] initiate the studies in this field by presenting the HowTo100M dataset. It is composed of 1.2 million instructional YouTube videos demonstrating thousands of different tutorials about cooking, crafting, gardening, repairing, and such. The narrations are provided as subtitles generated by the Automatic Speech Recognition (ASR) system. The presented model uses pre-trained models' video and text features and projects them into a joint embedding space such that the video-subtitle pair cosine similarity score is high. The model is trained with contrastive loss.

Some works use the BERT [40] pre-training language model in video-text learning. VideoBERT [35] uses video clips as a sequence of visual words to be trained along with the textual model. Then, ActBERT [41] further inserts local and global tokens as actions and objects to add the learning of human-object interactions. ClipBERT [42] suggests using

fewer video frames to decrease the enormous computation drawback of video-text learning models by presenting state-of-the-art results. In addition, Yang et al. [43] propose using a different noun/verb aware contrastive loss to emphasize the object or action tokens in the learning. Then, Lin et al. [44] present SwinBERT in order to enhance long-range video modeling. It learns attention masks to extract sparse video tokens in an end-to-end manner along with a multi-modal transformer.

Due to misaligned video-subtitle pairs, Miech et al. [2] propose MIL-NCE loss. Here, for a given video, not only the related transcription is used but also surrounding transcripts are contributed to learning. Moreover, end-to-end learning is provided. In our study, we utilize the representations obtained from this model and a detailed description of the process can be found in Section 3.. Likewise, Tang et al. [45] attack the misaligned and incomplete subtitle problem. They propose adding dense captions to elevate the inadequate text information and using attention loss to focus on the most related caption among the others.

Following the success of MIL-NCE, COOT [37] further improves embedding space by using transformers on top of the MIL-NCE encodings. Patrick et al. [46] enhances the contrastive loss by proposing a generative loss that leverages support set captions in order to avoid strict discrimination of non-paired but semantically similar captions to the video. They use transformers for the encodings. In another study, Liu et al. [47] propose the Hierarchical Transformer (HiT) that utilizes both feature level and semantic level contrastive matching to leverage different levels of information in video-text learning.

Furthermore, audio is used as the third component in cross-modal learning models to improve the joint learning space. First, Gabeur et al. [48] propose a multi-modal transformer that encodes video, speech, and audio and jointly learns embeddings with related caption encodings from BERT. This transformer module provides temporal information for the occurrence of the action with the audio. Then, [49] suggest a multi-modal contrastive loss and [50] utilize transformers similar to [48]. Differently from previous works using audio as an additional component, in [51], they utilize static images as snapshots of the videos. They propose a dual transformer encoder that leverages spatial information from the

images and gradually learns temporal information from videos. Most recently, Shvetsova et al. [52] present combinatorial loss on the training of multi-modal fusion transformer. Gabeur et al. [53] propose exploiting the cues in the different modalities (audio, transcription, appearance) to a shared encoder through mask training in order to leverage joint learning of all information.

With the advance of the Contrastive Language-Image Pre-Training (CLIP) [39] model, Luo et al. [19] propose using it for video-text learning. They transfer image representation to video representation by using a visual transformer and encoding positions of the frames along with their spatial features. Then, CLIP trains the video-text similarity learning scheme using related visual and textual sequences.

Moreover, Croitoru et al. [54] suggest a teacher-student method using distillation loss with multiple text embeddings. This is also shown to be useful for noise elimination in datasets. Li et al. [55] propose improved multi-modal learning using video patches to learn semantic similarities between objects and related text. They present a new task where the video-text model predicts entities to reinforce multi-modality in addition to regular video and text encoders. Similarly, Wang et al. [56] incorporate object bounding boxes into the visual transformer training for an object-aware space.

2.2. Action Localization on Instructional Videos

Localizing actions in instructional videos requires detecting action steps in the given video by weak supervision. For this purpose, many labeled datasets have been introduced to evaluate action localization where they have action step lists or descriptions for each instruction [27, 30, 33, 36, 57–59]. The task is to find the best alignment between the encodings of given action steps and short clips of the videos. Therefore, the video-text representation model impacts the process. The alignment is accomplished by Viterbi, Hidden Markov Model (HMM), and Dynamic Time Warping (DTW) kind of methods in order to maximize the total alignment of observations. We detail the related studies in the following.

First, Richard et al. [60] examine weakly supervised segmentation of fine-grained actions. They do not use any text encodings but uniformly split a video into the number of steps to appear according to the related instruction and then iteratively train RNN to finally determine regions of the actions. It is followed by [61] where they utilize a Viterbi-based loss and then they present an extended HMM formulation [62].

Zhukov et al. [33] propose to utilize narrations in the videos to localize and learn models for each cross-task action. This is conducted by applying an alternate update between the action's location and the parameters of the action classifier. After learning models with weak supervision from narrations, they run each model on temporal segments and extract a scoring matrix for video clips and action step models. The best alignment is found by maximizing the alignment scores with dynamic programming. Here, each action can only be assigned to one video clip. In our study, we utilize this same inference methodology which is also used in [2]. Furthermore, Lin et al. [63] propose matching textual instructions in wikiHow with the generated transcriptions in order to label instructional action steps automatically. Then, they show the usage of this supervision by training video models and presenting results on downstream tasks.

Weakly-supervised video to action step alignment, where only the ordered action steps are available for training, is explored well by improving the traditional sequence-to-sequence alignment method DTW either with discriminative and differentiable DTW [64], constraint alignment [65], restricted edit distance [66], segment-level beam search [67] or pairwise ordering consistency [68]. Then, Dvornik et al. [69] further propose Drop-DTW, which suggests a modification for DTW to solve assigning outliers to action steps. Similar to our motivation, the goal is to decrease the adverse effect of misleading outlier scores. Differently, they focus on solving this issue from the prediction part directly. Likewise, they conduct experiments on CrossTask and COIN using MIL-NCE [2] representations and present improved results. Han et al. [70] propose the Temporal Alignment Network (TAN) that encodes the alignability of the video and text inputs to mitigate the non-aligned and noisy video-text pairs' adverse effects on DTW. Moreover, in [71], they present an approach to detect whether the corresponding narration is visually described in the pairing video

clip. To this end, they use a small partially labeled dataset composed from COIN [36] and CrossTask [33] to train their joint model and then assign pseudo labels on large-scale HowTo100M [34] dataset to cope with the noisy training pairs of it.

Besides weakly supervised action localization, many studies propose unsupervised segmentation of the actions in instructional videos. Alayrac et al. [31] utilize clustering of video and narration segments. Sener et al. [72] suggest an iterative discriminative-generative approach through the Generalized Mallows Model (GMM) for segmenting video clips. Kukleva et al. [73] introduce clustering time-stamp based embeddings to obtain sub-action representations and encode videos accordingly. Furthermore, Elhamifar et al. [74] present a key-step extraction module along with the subsequential studies [75, 76] proposing the usage of narrations.

2.3. Actionness Learning

Actionness learning is widely studied to improve the Weakly Supervised Temporal Action Localization (WS-TAL) task applied to fundamental activity videos. The ultimate goal is to detect time-stamps of the given single-type activities, such as throwing a ball or running, without knowing the locations of the actions. Thus, studies leverage the weak supervision of video-level labels to classify uniformly sampled temporal proposals with attention mechanisms [77] and obtain class activation sequences. After that, the aim is to filter out noisy activations by learning the actionness of the segments. While these videos present a comparably easier task due to short and basic scenes, applying the same process to instructional videos is challenging because multiple actions appear in a complex manner. Therefore, other solutions, such as using pre-defined action steps or unsupervised localization of different entities are explored, as previously mentioned. As far as we know, only Zhukov et al. [1] attack the problem of actionness learning to improve action localization on instructional videos. However, the actionness score is used for post-processing action localization results, unlike studies presented below for WS-TAL, where actionness loss contributes directly to model learning. We attack the problem as the same as [1] and focus

more on discriminating actions from backgrounds. In the following, we comprehensively review the actionness learning literature for WS-TAL to present the state-of-the-art methods that could be applied to instructional videos in the future.

Actionness is first explored as defining the likelihood of frame regions [78] or motion maps [79] for containing an action scene, and it is shown to be useful for action detection. After that, studies focus on the discrimination of background frames that does not include any action scene to denoise the false activations obtained from action detection. To this end, explicit modeling of backgrounds is proposed for temporal action localization in addition to modeling of actions [80, 81].

First, Nguyen et al. [77] introduce a background-aware loss that contributes to the total training loss along with clustering and action losses, and they show an improvement compared to only using action loss. Lee et al. [82] propose a background suppression branch to avoid misclassifying backgrounds into action classes. They suggest using an auxiliary class for backgrounds to filter their activity scores. Ma et al. [83] use actionness loss jointly with frame-wise classification scores, which are supervised with single-frame annotations. Similarly, Lee et al. [84] use single-frame annotations to integrate background and action scores for joint learning. Furthermore, Narayan et al. [85] combine action/background discriminating loss with denoising loss to have enhanced action detection scores. Chang et al. [86] propose a transformer module augmented with actionness prediction to be used along with an adaptive graph to detect noisy actions and learn temporal context jointly.

In [87], they suggest separating scenes containing real action from scenes that do not include action but bring confusion due to contextual similarities. It is accomplished by first getting snippet level predictions from attention-based video classification model [88] and then learning two representation spaces jointly with triplet loss: (1) Contextually similar samples close to background scenes and distinct from action scenes, (2) contextually similar samples close to action scenes and distinct from background scenes. Likewise, Ma et al. [89] attack the problem of misclassification due to similar context appearance in the scenes and propose training an actionness model that separates actions from context.

2.4. Image to Video Adaptation

Web images are widely utilized in understanding human activities either for compensating the challenges in labeling video data or supporting the localization of actions. To this end, separate action models are trained by using action images [90]. Many studies focus on improving the mining of noisy web images [91, 92], decreasing the domain gap between images and videos [93, 94], or training models jointly with images and videos [38], similar to our experiments in Section 5..

Sun et al. [95] propose a domain transfer strategy in order to filter noisy web images and weakly labeled non-action video frames jointly. Then, models are trained over cleaned images for assigning action scores to video frames before learning action detectors. Similarly, Zhang et al. [96] attack the domain shift problem. However, they propose combining image and video features and then optimizing image and heterogeneous feature classifiers jointly.

Furthermore, Yu et al. [97] suggest using symmetric GANs (Generative Adversarial Network) to obtain domain-invariant representations for images and videos where one GAN maps joint features into video space and the other one maps video features into joint feature space. Liu et al. [98] introduce the inclusion of keyframes along with image and video models to guide the training of domain-invariant feature space.

Duan et al. [99] transform web images into pseudo videos by inflating data before learning joint space, and they use teachers to distill noisy web data. Kae et al. [100] suggest training spatial and temporal features sequentially by a two-stage approach rather than joint training. They first learn an image model and use the parameters to initialize the video model to be trained later. Chen et al. [101] propose a spatial-temporal causal graph approach to solve the domain shift problem. Most recently, Lin et al. [102] present a domain adaptation method that alternates between updating the image model with pseudo labels obtained from the video model and visa versa.

3. ACTION LOCALIZATION AND ACTION/BACKGROUND DISCRIMINATION ON INSTRUCTIONAL VIDEOS

This chapter presents how the action localization task is done on instructional videos using the joint video-text embedding space proposed with MIL-NCE [2] model and what can be done to improve the process. Firstly, the baseline model MIL-NCE is explored, and then the action assignment process done on video clips of the CrossTask [33] dataset using pre-defined action step labels is explained. After that, the action localization results on the validation set of CrossTask are analyzed to identify opportunities for improving step recall. Our investigations indicate that mitigating the detrimental effects of background video clip scores could improve the step recall results. To this end, we examine how to represent backgrounds and actions so that by discriminating them, we can post-process the scores accordingly before doing the step assignment. In other words, we aim to learn the *actionness* of a video clip, similar to previous research [1].

In this regard, we explore the possibility of representing video clips using cosine similarity scores between pre-defined action step labels. We claim that actions and backgrounds have different patterns regarding their similarity score distribution among the action step labels and neighbor video clips. For example, backgrounds might get equally distributed low scores for all action step labels, whereas action clips might get a high score for a specific action step label. Thus, a video clip can be defined by its similarity scores. Finally, we show how we learn a discrimination model between actions and backgrounds with the novel representations.

3.1. Action Localization on CrossTask with MIL-NCE

3.1.1. CrossTask

The CrossTask dataset, introduced in Zhukov et al. [33], is a collection of 2750 instructional videos from 18 primary tasks, which are annotated with pre-defined ordered action step labels. For example, the cooking task "Make Banana Ice Cream" has action step labels as "peel banana, cut banana, put bananas into blender, pour milk, mix ingredients". The videos are labeled with varying duration and order according to the appearance of the actions. In [33], 360 annotated videos (20 videos for each task) are held as the validation set and used to tune the hyperparameters of the models. We divide this set into val-train, val-holdout and val-test (12, 4 and 4 videos for each task, respectively). We aim to work on a small annotated subset and not touch the original test sets except for computing the final result. As in [33], we randomly obtain original test sets from the remaining 2390 videos 20 times and report the average.

3.1.2. MIL-NCE

The MIL-NCE proposed by Miech et al. [2] is a jointly trained end-to-end model learned for obtaining robust video representations with the aid of narrations rather than manual annotations. This model is trained on the HowTo100M [34] dataset, which consists of 1.2 million instructional YouTube videos. The narrations in the videos are provided as subtitles generated by the Automatic Speech Recognition (ASR) system. A total of 120 million video clips are obtained based on the appearing intervals of the subtitles provided by YouTube. The data is split into 3.2 second clip-text pairs.

However, as it is told in Miech et al. [34], the alignment between subtitles and demonstrations in the HowTo100M dataset is often inaccurate. The reason is that narrators may explain the steps before or after performing them. This leads to misleading similarity learning results between video clips and subtitles. To address this issue, the authors propose not only

considering the subtitle that matches the video based on time intervals but also taking into account the subtitles before and after. This method involves selecting the closest subtitle to the video from a bag of subtitles with a Multiple Instance Learning (MIL) approach. On the other hand, coarsely, the contrastive loss function, Negative Contrastive Estimation (NCE), ensures that the similarity between a positive video and a negative subtitle (i.e., a subtitle that does not belong to that video) is smaller than the similarity between the positive video and its own subtitle. This approach results in matched subtitles and video clips being much closer to each other in the learned joint space than in the negative set. Consequently, they combine Multiple Instance Learning (MIL) with Negative Contrastive Estimation (NCE) to develop a robust model that can handle misaligned video clip-text pairs.

The model architecture taken from [2] is presented in Figure 3.1. The architecture employs two distinct gates for processing video clips and related text. The video gate f leverages either the I3D [10] or S3D [103] models as backbones and we use S3D as the video encoder throughout our study. The text gate g processes each subtitle by breaking it down into individual words and representing them using their corresponding word2vec representations. The projection layers serve as the final linear layers for embedding both modalities into a shared space. The model is trained from scratch in an end-to-end manner. The similarity between a video clip x and its corresponding text y is computed using the dot product of their embeddings, $f(x)g(y)^T$, for downstream tasks such as action localization on CrossTask.

3.1.3. Action Localization on CrossTask

We present the methodology employed for action localization on CrossTask using video and text encodings obtained from our baseline model, MIL-NCE. Initially, each video is segmented into one-second clips, and text encodings for a primary instructional task's action step labels are acquired. The similarity scores are computed between each clip and pre-defined action step label encodings, resulting in a cosine similarity table. Each row comprises a video clip's action step similarity scores. The best video clip for each action step is identified using dynamic programming to solve Equation 1, taken from Zhukov et al. [1].

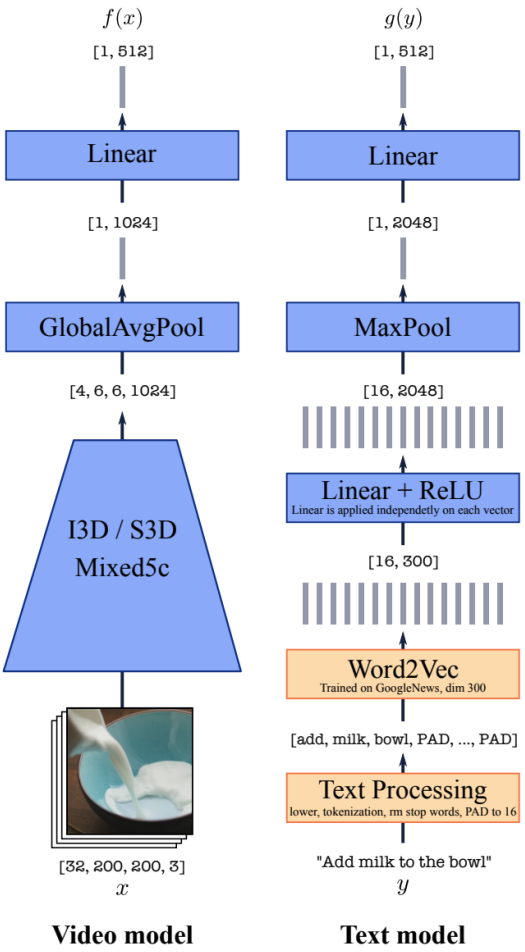

Figure 3.1 The baseline model trained with the MIL-NCE loss. This figure is taken from [2].

Here, $f_k(x_t)$ denotes the similarity score between video clip x_t and action step k, while $t_1, ..., t_K$ represent the clip ids assigned to steps $1, ..., K$, respectively. An order constraint exists where the latter video clips are assigned to the last steps, and the total similarity is maximized accordingly. Evaluation is done by computing the "step recall", defined as the ratio of correctly predicted video clips to the total number of action steps annotated.

$$(t_1^*, ..., t_K^*) = \arg \max_{t_1 < ... < t_K} \sum_{t=1}^{T} \sum_{k=1}^{K} f_k(x_t) \tag{1}$$

However, as seen from the statistics in the Section C.2.-Table 2[2] of [33], three problems may mislead the action step assignment maximization. First, a considerable percentage of

[2]https://arxiv.org/pdf/1903.08225.pdf

action steps are missing in the videos, averaging 31%. This leads to the dynamic prediction process forcing the prediction of unseen actions, resulting in false positive predictions on backgrounds or video clips of other action steps. Second, false positive predictions may also arise due to the order constraint, given that the action steps do not always appear in the given order due to the nature of the web videos. The order consistency is given as 86%, while it is relatively lower for some tasks. Lastly, the large proportion of background frames forming an average of 72% of videos can adversely affect the similarity maximization process, resulting in false positive predictions on backgrounds due to the misleading action step similarity scores. While these problems can be solved by improving the prediction method as in [69], handling challenging background clip scores depends on also the robustness of the baseline video-text model. In this study, we focus on enhancing the recognition of backgrounds by arranging the similarity scores accordingly without changing the baseline model or retraining it.

Specifically, we identify the occurrence of false positives when a background behaves as an outlier that cannot be distinguished from action scenes by the baseline model. We then concentrate on this issue and further examine it. Firstly, we demonstrate that challenging background similarity scores have an important impact on the average step recall results. We employ val-test labels and manipulate the scores of background video clips by setting them to -inf, indicating the lowest possible scores. This approach allows us to compute an upper bound. The comparison of the baseline and the upper bound task-based step recall results are given in Table 3.1. Here, the average baseline step recall is **42.35**, while the upper bound for background score modification is **55.47**. Moreover, there is a significant improvement in the majority of the tasks.

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
MIL-NCE [2]	45.00	37.50	46.66	28.00	62.50	52.38	36.58	58.33	33.33	37.50	35.29	9.09	38.23	35.29	37.50	69.23	53.17	44.44	42.35
Upper bound	65.00	43.75	73.33	52.00	62.50	42.85	56.09	58.33	50.00	50.00	58.82	36.36	47.05	52.94	54.16	73.07	77.77	44.44	55.47

Table 3.1 The baseline and upper bound step recall results for each task when background clips get the lowest scores. The evaluation set is val-test.

This promising outcome suggests that modifying the baseline scores to improve the differentiation between action and background clips has the potential to enhance action localization. To automatically achieve this, we need to learn the discrimination of actions and backgrounds according to their behaviors. We propose to learn this discrimination from the action step similarity score cues, which we present in Section 3.2..

3.2. Action/Background Discrimination with Similarity Cues

In this section, we utilize the val-train set containing 216 videos. First, for each video clip x, we calculate cosine similarity scores between all 105 action step labels $k \in K_D$ where k is an action step label and K refers to all action step labels in the dataset D, CrossTask. Then, we apply the softmax function to have a $1\text{x}105$ probability distribution a_x for each video clip x. An overview of the approach is given in Figure 3.2. Each row corresponds to a_x, and we call them as *"video clip to action step label assignment probabilities"* (video-to-action probability). We aim to define each video clip by different cues from probability scores and then investigate whether they contribute to the discrimination.

Our approach involves the development of four main components. Firstly, we assume that action video clips tend to have higher scores for specific action steps, while backgrounds tend to have lower scores even if they are assigned to a step. To capture this, we define the first component, r_1, as the maximum probability score from a_x for each video clip x. Secondly, we utilize the entropy of a probability vector to gain insight into the distribution of scores. Action video clips typically exhibit lower entropy due to having more confident scores on specific action steps, while background video clip scores tend to show high aleatoric uncertainty and therefore have higher entropy values. We define the entropy values of each video clip as r_2. Thirdly, we create a $1\text{x}3$ vector from the neighbor values of the action step with the maximum probability value, as shown in Figure 3.2-r_3, and calculate its entropy. Our intuition is that backgrounds tend to be outliers in the region, which results in a lower r_3 value for backgrounds and a higher value for action clips since actions appear sequentially. Lastly, we calculate the average of the regional entropy as the fourth component, r_4. We

compute r_2 for surrounding neighbors and the current video clip x, as shown in Figure 3.2-r_4, and then take the average of the three values. This is based on the idea that backgrounds and actions may appear together in a region, and therefore, the average entropy can be representative. We combine all four components to form the $\mathbf{r_x}$ vector, which represents each video clip by its action step label assignment probability cues.

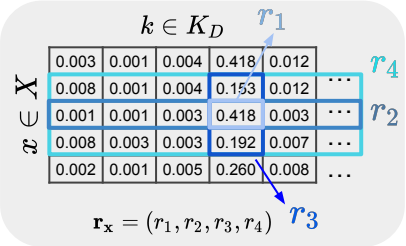

Figure 3.2 Components calculated from the video clip $x \in X$ to action step label $k \in K_D$ assignment probability cues. Final vector is formed as $\mathbf{r_x}$.

In order to investigate the capability of $\mathbf{r_x}$ to represent actions and backgrounds, we employ the scikit-learn [104] LinearSVC implementation to learn a linear Support Vector Machine (SVM), with $C = 100$ and balanced class weights. Specifically, each video clip x has a target y, where $y = 0$ represents backgrounds and $y = 1$ represents action clips. Inputs are scaled with zero-mean unit-variance. Following the training of the SVM model, we analyze the feature weights to gain insight into the most effective component. Figure 3.3 indicates that calculated entropy values and max probability values serve as reliable discriminators. However, we observe that the small vector size of r_3 results in reduced contribution to the classification performance. Moreover, extending the window size to obtain more neighboring frames is found to be detrimental, due to the increased presence of misinformation resulting from the action clips appearing at the start or end of the action window. Finally, we evaluate the model on the val-test set and report the average precision of action/background discrimination as **36.48%**.

3.3. Discussion

Some background video clips' action step label similarity scores can result in misleading information due to the sensitivity of video-text embedding models. The performance can be

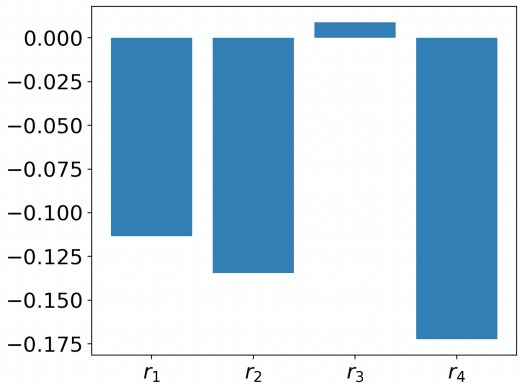

Figure 3.3 Linear SVM weights assigned to the components of $\mathbf{r_x}$.

enhanced by recognizing and decreasing these scores, as indicated by upper bound results. Our suggestion is to utilize the cues obtained from similarity scores, as background and action video clip scores can reveal representative patterns. To this end, we propose converting each video clip's similarity score vector into a probability distribution.

The probability cues alone are not enough to have a sufficient discrimination model that can be used to augment similarity scores since the average precision is low. Therefore, after showing the video-to-action probability cues can contribute to action/background discrimination, we continue to develop our approach in order to improve the baseline MIL-NCE visual features for better discrimination. By having that, our final goal is to post-process the similarity scores according to a video clip's probability of being an action or background to decrease the adverse effect of challenging background scores. A similar approach is presented in Zhukov et al. [1] where they augment the similarity scores with their *actionness* model that gives probabilities for being an action or background. We further propose a model by combining visual features and action step label assignment probability cues.

In our previous work [4], we train a discrimination model jointly trained with MIL-NCE visual features and an auxiliary feature that we form based on the action step label assignment probability cues. We propose to use an extended version of the 4-dimensional representation $\mathbf{r_x}$ and show the improvement of it over visual features in ablation studies. We further explain this study in the next chapter.

4. LEARNING ACTIONNESS FROM ACTION/BACKGROUND DISCRIMINATION[3]

As discussed in the previous chapter, despite the promising achievements of video-text joint embedding space models, one of the essential issues that affect their robustness is the background video clips, not including any action related to the task in the video. Some background scenes can be difficult to differentiate from action scenes due to similar words in narrations or objects. As a result, action localization is adversely impacted, and identifying these misleading background scenes can improve performance, as shown in Table 3.1.

One approach to address this issue is to learn the **actionness** of a video clip, which determines the probability that it contains an action. Zhukov et al. [1] propose a self-supervised temporal order verification method to learn actionness. This approach enhances action localization by increasing the probabilities of action clips and decreasing those of background clips to avoid false predictions. With similar objectives, we introduced a new approach for learning actionness that aims to reduce false positive predictions on backgrounds in our previous work [4]. Specifically, we train a binary logistic regression model to discriminate between action and background clips.

We incorporate video-to-action probability cues as an auxiliary representation in conjunction with visual features to enhance classification performance, as demonstrated in Section 3.2.. Our rationale is that background video clips tend to yield similar scores for all action step labels, while action clips exhibit more robust scores for specific steps. Furthermore, since action clips frequently appear in sequence, they exhibit the highest scores for similar steps. Consequently, we leverage the cues from adjacent video clips. By jointly training the model, we aim to project visual features onto a space that is more sensitive to action/background discrimination. During inference, we employ the actionness score of each video clip to adjust the video-to-action outputs of the baseline methods through post-processing.

[3]This chapter is adapted from our Signal, Image and Video Processing'22 journal paper [4].

The methodology employed in this study is demonstrated in Figure 4.1, which depicts the various stages involved. It is noteworthy that, as per the baseline scores, the "pour water" action step label has been erroneously assigned to a background scene. However, through the application of actionness scores to each video clip in post-processing, we have successfully rectified this issue.

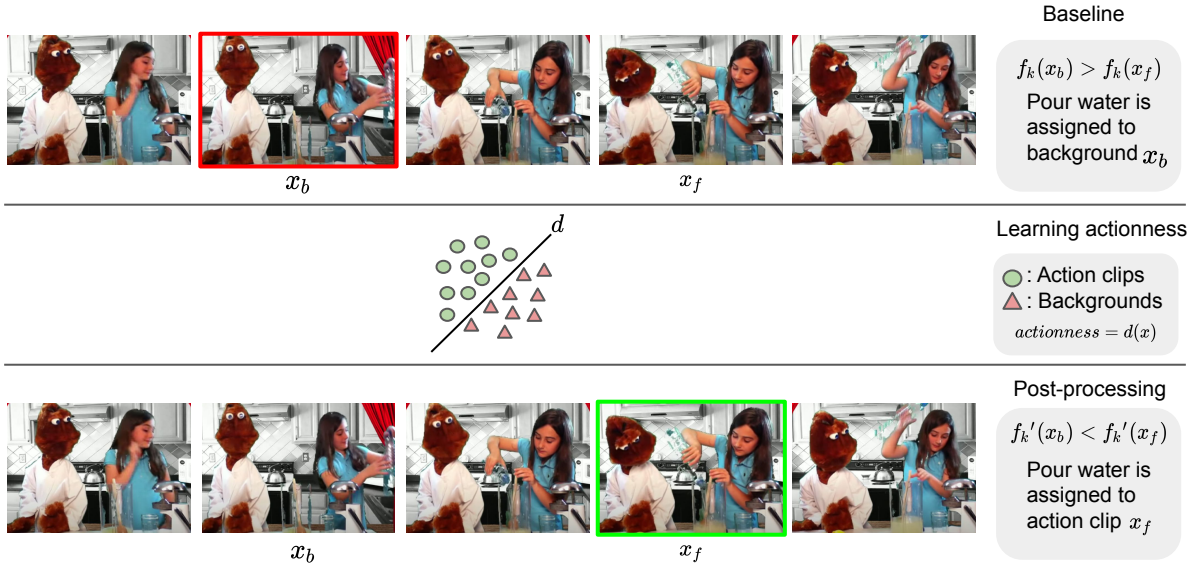

Figure 4.1 Overview of our method. We learn a discrimination model d to derive an actionness score $d(x)$, indicating the probability of an action being present in the given input x. Leveraging the output of the baseline video-to-action outputs $f_k(x)$ for each video clip x and action step label k, we integrate the actionness score with the baseline scores to obtain the post-processed scores $f_k^{'}(x)$. Specifically, the proposed approach corrects false positive predictions on background clips, such as the "pour water" action step in the illustrative example, and elevates the scores of action clips to identify the associated action step label accurately.

The subsequent sections outline our approach for learning the actionness of video clips to augment the baseline methods. We test our method on both the CrossTask and COIN datasets. We present that the combination of actionness scores with baseline video-to-action outputs yields enhanced action step localization outcomes. Furthermore, we show the results of action segmentation as an additional task. Ablation studies are conducted to provide insight into our method and the selected parameters.

4.1. Method

Our study aims to improve the accuracy of action step labeling for a given video clip x. We first train a binary discrimination model d to classify action and background video clips and obtain the actionness score $d(x)$ for the given clip x. In the second step, we use the actionness score to refine the baseline model outputs for the video-to-action outputs denoted by $f_k(x)$, where k represents the pre-defined action step label. We apply the actionness score to post-process the baseline outputs, resulting in improved step assignments $f_k^{'}(x)$. Our ultimate goal is to predict better action step labels for each video clip x based on the post-processed outputs.

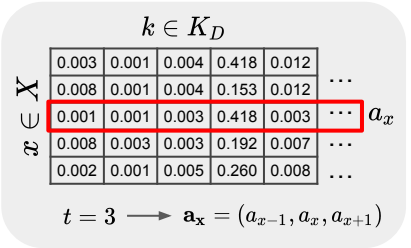

Figure 4.2 Proposed auxiliary representation denoted as $\mathbf{a_x}$ for a video clip x. The representation is constructed using the probabilities $f_k(x)$ of assigning action step labels to a video clip x. The temporal window for selecting neighboring probabilities is set to $t = 3$.

4.1.1. Learning Actionness

We propose a method to improve the discrimination of video clip representations by incorporating auxiliary features. Each video clip x is represented by a visual feature $\mathbf{v_x}$ and an auxiliary feature $\mathbf{a_x} \in \mathbb{R}^{K_D^t}$, which is computed only for training samples. The auxiliary feature is constructed by considering the probability of each action step k in the related dataset D, within a temporal window of size t that includes the current clip x and its neighbors (Figure 4.2). The proposed vector includes baseline video-to-action probabilities $f_k(x)$, and the neighbors are concatenated end to end to form the final auxiliary feature $\mathbf{a_x} = (a_{x-\lfloor t/2 \rfloor}, ..., a_{x+\lfloor t/2 \rfloor})$. The size of $\mathbf{a_x}$ is $\mathbb{R}^{105*t}$ for CrossTask and $\mathbb{R}^{779*t}$ for COIN

dataset, and we set $t = 3$ for our experiments. We consider two types of visual features: the global average pool outputs of the video model in MIL-NCE S3D [2] with $\mathbf{v_x} \in \mathbb{R}^{1024}$ and the intermediate low-level clip embedding of COOT [37] trained on YouCook2 with $\mathbf{v_x} \in \mathbb{R}^{384}$. Each video clip x is associated with an annotated target label y, where $y \in 0, 1$, with 0 indicating backgrounds and 1 indicating actions.

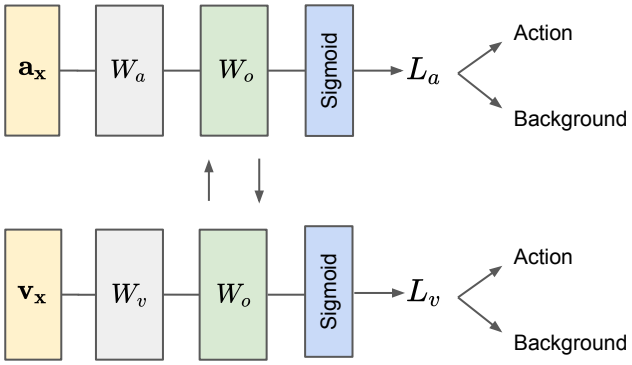

Figure 4.3 The proposed discrimination model d to obtain the actionness of a video clip x. It is jointly trained with auxiliary $\mathbf{a_x}$ and visual $\mathbf{v_x}$ features. The actionness $d(x)$ is the probability of being an action for the visual representation $\mathbf{v_x}$.

Incorporating the auxiliary representation, we enhance the discrimination model d by training it with shared weights for the input vectors $\mathbf{v_x}$ and $\mathbf{a_x}$. The model architecture, as illustrated in Fig. 4.3, involves projecting the inputs onto a fixed dimension of $\mathbb{R}^{512}$ using projection weights W_v and W_a. After that, the shared output linear layer W_o is utilized to classify the inputs, followed by sigmoid functions and a joint loss:

$$L = L_v + \theta * L_a \tag{2}$$

The hyper-parameter θ balances the contribution of the auxiliary vector loss and is assigned a value of 0.8. The visual vector and the auxiliary vector classification loss functions denoted by L_v and L_a, respectively, are the binary cross-entropy losses used for distinguishing actions from backgrounds where

$$
\begin{aligned}
L_v &= -\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(\mathbf{v}_{\mathbf{x_i}}^{'})) + (1-y_i).log(1-p(\mathbf{v}_{\mathbf{x_i}}^{'})) \\
L_a &= -\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(\mathbf{a}_{\mathbf{x_i}}^{'})) + (1-y_i).log(1-p(\mathbf{a}_{\mathbf{x_i}}^{'}))
\end{aligned} \tag{3}
$$

and

$$
\begin{aligned}
p(\mathbf{v}_{\mathbf{x}}^{'}) &= P(y=1 \mid \mathbf{v}_{\mathbf{x}}^{'}) = \frac{1}{1+e^{-(W_o^T \mathbf{v}_{\mathbf{x}}^{'}+b)}} \\
p(\mathbf{a}_{\mathbf{x}}^{'}) &= P(y=1 \mid \mathbf{a}_{\mathbf{x}}^{'}) = \frac{1}{1+e^{-(W_o^T \mathbf{a}_{\mathbf{x}}^{'}+b)}}
\end{aligned} \tag{4}
$$

Here, $\mathbf{v}_{\mathbf{x}}^{'} = W_v^T \mathbf{v}_{\mathbf{x}}$ is the projected visual feature and $\mathbf{a}_{\mathbf{x}}^{'} = W_a^T \mathbf{a}_{\mathbf{x}}$ is the projected auxiliary feature of video clip x. W_o and b are the learned weight and bias, respectively. $P(y=1 \mid \mathbf{v}_{\mathbf{x}}^{'})$ or $P(y=1 \mid \mathbf{a}_{\mathbf{x}}^{'})$ depicts the binary model prediction probability for whether x belongs to the action class.

In inference time, we obtain the actionness score as

$$
d(x) = P(y=1 \mid \mathbf{v}_{\mathbf{x}}^{'}) = \frac{1}{1+e^{-(W_o^T \mathbf{v}_{\mathbf{x}}^{'}+b)}} \tag{5}
$$

We only use visual features in inference time in order to decrease the complexity of calculating auxiliary representations for test sets and provide a generalized pipeline for our method to be adapted to all video datasets in the future. Since labeling or providing action step lists for instructional videos cause additional effort, we limit inference to visual features. Thus, auxiliary representation obtained from video-text similarities is required only in the training time for a small labeled set.

4.1.2. Post-processing with Actionness

In order to get baseline video-to-action outputs $f_k(x)$, we utilize two datasets and models which indicate different downstream tasks. We modify the baseline model outputs with the

proposed actionness scores as follows:

$$f_k^{'}(x) = (1-\alpha)f_k(x) + \alpha d(x) \tag{6}$$

We utilize the CrossTask [33] dataset to perform **action step localization**. We employ the MIL-NCE S3D model to obtain baseline scores $f_k(x)$ by utilizing the last layer's output, as previously demonstrated in [2]. This involves acquiring 512-dimensional representations for each video clip from the video model and each action step label k in primary tasks of CrossTask from the text model. To determine the cosine similarity scores $S(X, K_{CT})$ between a video clip $x \in X$ and each action step label k, where X denotes the test set video and K_{CT} comprises all step labels in the CrossTask dataset, we utilize the MIL-NCE representations. The resulting $f_k(x)$ is set to $S(x, k)$ for action step localization. Before generating the auxiliary feature $\mathbf{a_x}$, we subject $S(x, K_{CT})$ to the softmax function to derive probabilities. Finally, we combine the actionness scores $d(x)$ of each video clip x as

$$S^{'}(x, K_{CT}) = (1-\alpha)S(x, K_{CT}) + \alpha d(x) \tag{7}$$

to obtain post-processed outputs $f_k^{'}(x) = S^{'}(x, k)$, and then we re-predict video-to-action assignments as done in [1].

We use the COIN[36] dataset for **action segmentation**, along with the representations from two baseline models. Moreover, we propose a novel post-processing method for this task. We first obtain 2112-dimensional features from the MIL-NCE model as described in [2] and 384-dimensional features from the COOT model as in [37]. To establish the baseline video-to-action probabilities, a multinomial logistic regression c is trained on the training set of COIN. Here, each action step label, along with the background, is considered as a class $k \in K_C$, where K_C has 779 targets. Softmax probabilities for each x are obtained as $f_k(x) = c(x) = P_c(y = k \mid x)$, where y represents the class label. In contrast to the previous task, where the probabilities were updated for all action steps, for this task, the approach is different. Since there is a probability of being a background $k = 0$, the score of the action

step with the maximum assignment probability $k = m$ is considered the clip's main action step score in addition to $k = 0$. As such, the clip's background probability $P_c(y = 0 \mid x)$ is combined with the probability of being a background $1 - d(x)$ and the clip's main action probability $P_c(y = m \mid x)$ is combined with the probability of being an action (actionness) $d(x)$:

$$
\begin{aligned}
P_c^{'}(y = m \mid x) &= (1 - \beta)P_c(y = m \mid x) + \beta d(x) \\
P_c^{'}(y = 0 \mid x) &= (1 - \beta)P_c(y = 0 \mid x) + \beta(1 - d(x))
\end{aligned} \tag{8}
$$

$f_{k=0}^{'}(x) = P_c^{'}(y = 0 \mid x)$ and $f_{k=m}^{'}(x) = P_c^{'}(y = m \mid x)$ are the post-process assignment probabilities for background and main action step but we leave other ones unchanged. In our experiments, both α and β are weighted combination parameters, and we tune them on a holdout set.

4.2. Datasets

We use **CrossTask** [33] dataset for the action localization task and **COIN** [36] dataset for the action segmentation task. We give the details of CrossTask and the splits we utilize in Section 3.1.1.. As presented by Tang et al. [36], the COIN dataset consists of a comprehensive collection of instructional YouTube videos encompassing 180 distinct tasks. An ordered list of steps and corresponding annotations accompanies each video. The dataset is divided into two subsets, with 9030 and 2797 samples assigned for training and testing. Likewise, we randomly select 20 videos from each task within the training set to form a validation subset further partitioned into val-train, val-holdout, and val-test. Notably, our selection process ensures that all action step labels remain present within the remaining training set.

The data splits given in Table 4.1 are used as follows: We use val-train sets to train d models while using val-holdout sets to validate the training and tune the combination parameters. We

decide the model according to val-test results given in the ablation studies, and then present the final outcome with the original test sets.

	val-train	val-holdout	val-test
CrossTask	216	72	72
COIN	2160	720	720
Usage	Train d	Validate training, tune α/β	Ablation studies

Table 4.1 # of videos included and usage of different splits.

4.3. Implementation Details

In the MIL-NCE S3D [2] model employed in this study, video clip inputs consist of 16 frames sampled at 16 fps, resulting in one feature per second. The resolution of these frames is set to 200x200, and the 512-dimensional output of MIL-NCE serves as input to COOT [37]. To obtain the baseline MIL-NCE representation of COIN, we get the global average pool outputs of the video model in the dimension of 1024 for each video clip $x \in X$. Subsequently, we concatenate each video clip feature with the average representation of the video (1x1024) and sinusoidal position encodings (1x64) following [2], thereby resulting in a 1x2112 dimensional representation. The optimization procedure of the actionness model d entails using binary cross-entropy loss and ADAM optimizer with a learning rate of 10^{-5} and a batch size of 32. We train each model until the validation loss stops decreasing for 10 epochs, and saturation is achieved after 3-4 epochs. Furthermore, we scale the input embeddings to zero-mean unit-variance.

4.4. Experiments

4.4.1. Action Step Localization

We utilize the prediction and evaluation methodology outlined in previous works [1, 2, 33]. The evaluation metric employed is the average step recall, which measures the correctly

assigned action step labels in the CrossTask dataset. The details are given in Section 3.1.3.. To reproduce the step recall results reported in [2], we apply prediction on the test set similarity scores $S(X, K_{CT})$. Upon completion of the training process of d, we obtain actionness values for each video clip in the test set using Equation 5. We then incorporate these actionness values with each $S(x, K_{CT})$ via Equation 7 with the chosen value of $\alpha = 0.75$, following which we rerun dynamic prediction on S'.

Method	SR
MIL-NCE [2]	40.46*
Zhukov et al. [1]	41.00
Ours (d)	**41.76**

Table 4.2 **Action Step Localization.** Post-processing with actionness scores of the discrimination model d increases the step recall (SR) on CrossTask. The method in [1] improves the same baseline [2]. * depicts the reproduced baseline.

Zhukov et al. [1] propose a self-supervised method to learn action/background discrimination. On the other hand, we demonstrate in Table 4.2 that efficient actionness scores can be learned with simple binary logistic regression using a small amount of annotated data, which is jointly trained with auxiliary loss. The use of auxiliary representation enhances visual features as shown in Section 4.4.3., and our study outperforms [1] in improving over the same baseline [2]. Task-based results and the ratio of corrections from false positive predictions on backgrounds to true positives are presented in Table 4.3, where most classes show improvement. For the classes where we decrease the results, our method converts true positives to false positives as well. On average, we convert 17.03% of false positives on background video clips to true positives, but we have only a 1.3% improvement because we lose some of the baseline true positives too. This is because our discrimination model is not perfect, and post-processing can harm some classes where the model d is not robust enough to detect related actions. Qualitative results are also provided in Fig.4.4, showing examples of rectified assignments on backgrounds before and after post-processing. The proposed actionness score enables the correction of assignments into clips with actual action rather than only a scene with the object.

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry
MIL-NCE [2]	31.71	**40.80**	49.53	39.82	**24.90**	43.31	**36.21**	45.55	36.87	30.69	52.11	31.35	38.52	42.53	43.89	**47.84**	53.17	**39.44**
Ours (d)	**34.73**	38.66	**53.45**	**43.89**	24.20	**46.57**	34.21	**47.59**	**39.82**	**31.99**	**57.54**	**32.16**	**38.80**	**44.36**	**45.10**	47.15	**53.78**	37.80
Background FP to TP ratio	29.47	8.00	14.87	14.80	3.30	30.29	34.00	14.55	13.85	21.72	16.20	12.99	13.69	18.70	26.81	22.48	16.40	16.26

Table 4.3 Task-based CrossTask action step localization results (SR) and the ratio of corrections done from false positives on the backgrounds to true positives.

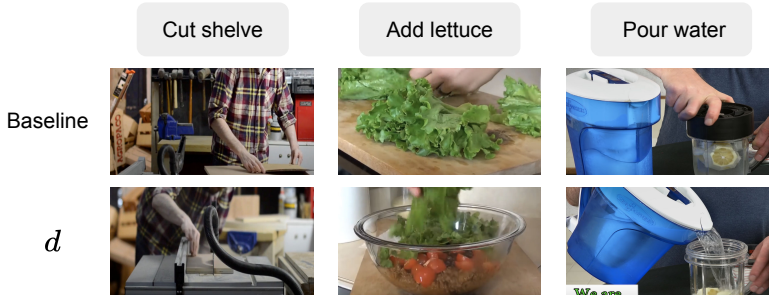

Figure 4.4 We present the predicted video clips for different action step labels before (baseline) and after post-processing with d. Our actionness score converts false positive predictions into true positives.

4.4.2. Action Segmentation

The COIN dataset's video clips are classified into an action step label or background class based on the maximum probability from c. The frame-wise accuracy is then calculated using the methodology outlined in [2]. The baseline model is trained and evaluated using MIL-NCE S3D or COOT features. We note that the discrepancy between the reproduced and reported MIL-NCE [2] baseline results from the random selection of the validation set. We train d on the val-train collection of COIN and obtain actionness values for the test set. We then combine d and c probabilities using Equation 8, where $\beta = 0.25$, before making step assignments on post-processed probabilities $P_c^{'}$. Based on these assignments, we recalculate the frame-wise accuracy and present minor improvements over two different baselines in Table 4.4. However, due to the high number of background scenes in COIN, this task remains open for improvement using alternative post-processing approaches to address the class imbalance problem.

Method	FA
MIL-NCE [2]	66.12*
Ours (d)	**66.34**
COOT [37]	63.72
Ours (d)	**63.84**

Table 4.4 **Action Segmentation.** Frame-wise accuracy (FA) results on COIN after applying post-processing with actionness scores of d. * depicts the reproduced baseline.

4.4.3. Ablation Studies

In this section, we investigate the impact of visual/auxiliary features and the number of data on the training of the discrimination model d. We present results on the val-test set of CrossTask.

4.4.3.1. Visual features: We conduct experiments to evaluate the impact of different visual features on the training of the actionness model d. The experimental results, which are summarized in Table 4.5, demonstrate the effects of these features on action/background discrimination, as measured by the average precision (AP), and post-processing, as measured by the step recall (SR). We observe that MIL-NCE features in the 1024-dimensional space produced the best discrimination performance. Consequently, we employ these features for subsequent tasks.

Feature	AP	SR
COOT 384 [37]	60.41	42.81
MIL-NCE 512 [2]	64.14	43.87
MIL-NCE 1024 [2]	**64.35**	**44.20**

Table 4.5 Various visual features are employed to train d with CrossTask val-test set. The average Precision (AP) metric measures the discrimination between actions and backgrounds, while the Step Recall (SR) is evaluated following post-processing with the trained d.

4.4.3.2. Auxiliary loss: We explore the impact of the auxiliary loss L_a, which corresponds to the binary classification loss of the auxiliary representation $\mathbf{a_x}$, in the joint

learning process of the model d with L_v. To achieve this, we conduct an experiment where we train the model solely with L_v, denoted as d_v in Table 4.6. Our findings demonstrate that incorporating the auxiliary loss enhances the visual features in the direction of improved discrimination between action and background video clips, leading to superior post-processing outcomes.

Model	AP	SR
Random	-	39.86
d_v	63.04	42.49
d_r	63.45	43.01
$d_{t=0}$	63.78	43.08
$d_{t=5}$	63.98	42.98
$d_{t=3}$	**64.35**	**44.20**

Table 4.6 Different configurations of the actionness model d and their impact on the performance metrics of average precision (AP) for action/background discrimination and step recall (SR). One such configuration involves employing random numbers generated from a uniform distribution as the actionness score.

We discuss the qualitative outcomes where the model d outperforms d_v. We illustrate these findings through Fig. 4.5, which displays action step samples predicted as true positives by d but false positives on backgrounds by d_v. Our study highlights that the joint loss function used to train d plays a crucial role in correcting false positive predictions and converting them into true positives. This is accomplished by incorporating an auxiliary loss that includes additional information about whether the scene is an action or background scene. The confidence score pattern observed in the baseline video-to-action probabilities provides a cue for practical model training. As a result, our research shows that learning weights for action step label scores provides improved discrimination performance compared to using only visual weights.

4.4.3.3. Neighbors in auxiliary representation: We further investigate the impact of neighboring video clips on the contribution of auxiliary features. To this end, we evaluate three jointly trained models using different auxiliary representations with varying dimensions. In Table 4.6, we present the results of our experiments, where $d_{t=0}$ denotes

Figure 4.5 Examples to where post-processing with d is better than d_v trained with only visual loss L_v.

the absence of neighboring video clips, $d_{t=5}$ refers to the use of a temporal window size of 5, and $d_{t=3}$ indicates a window size of 3. Our findings suggest that using too many or no neighboring video clips has a negligible impact on the auxiliary feature's contribution. In contrast, employing only the surrounding neighbors performs best when the window size is 3. This outcome is in line with our expectations, as the key actions tend to occur within a short period, and utilizing consistent cues enhances the quality of the information.

4.4.3.4. Using 4-dimensional auxiliary representation: We fix the dimension of the auxiliary feature to 4 by extracting further information from the probability cues. The details of this process are explained in the previous chapter (Section 3.2.). A model d_r is trained with the extracted features $\mathbf{r_x}$ as illustrated in Figure 3.2. However, our main model d, which is trained with $\mathbf{a_x}$, outperformed d_r as demonstrated in Table 4.6. This result suggests that learning weights from a pool of probabilities to extract cues provides more informative data than learning weights from hand-crafted values.

4.4.3.5. Amount of training samples: We explore the impact of the number of training videos utilized to learn the discrimination model d. We employ diverse val-train sets, each containing 1, 3, or 6 videos per task. Random videos from each task of CrossTask are selected, and each experiment is repeated 10 times to report the mean step recalls after post-processing. The results are presented in Figure 4.6, which displays the mean of the 10 runs for each val-train variant, along with the standard deviation with error bars. Our analysis

shows that increasing the number of videos does not significantly affect post-processing. Therefore, the amount of data in the training set is not a primary concern. However, as discussed in the subsequent chapter (Section 5.), the quality of the examples plays an essential role.

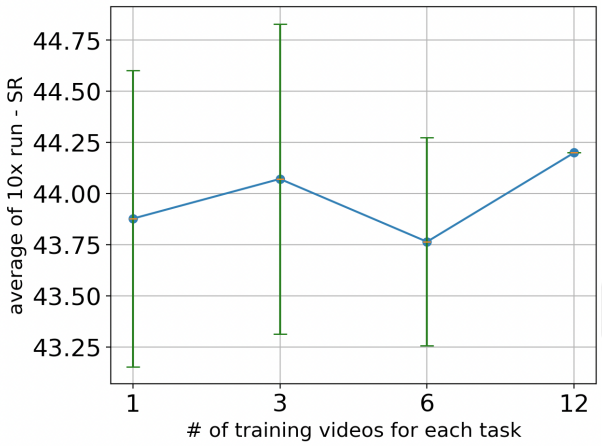

Figure 4.6 The effect of varying CrossTask val-train sets on post-processing. The baseline set contains 12 videos per task; its step recall is 44.20.

We fix the hyperparameters for all the ablation experiments as batch size, learning rate, and combination parameter α as the results are not changing significantly. Although training loss is decreased and the average precision of discrimination is increased a little more, we observe no difference in step recall results, which is in parallel with the above experiment. As we discuss later on in the following sections, the content of the training data affects our results. This is because we need weights that learn to discriminate a diverse set of actions rather than only a few actions. Therefore, even if our model's average precision and training are near perfect, if it has not learned all actions, our step recall would not be increased since the absent actions will not be detected.

4.5. Discussion

We introduce a novel approach to learning the actionness of video clips by leveraging the cues provided by the video to action step label assignment probabilities (video-to-action probabilities). To discriminate actions from backgrounds, we propose an auxiliary loss,

which improves the actionness scores and post-processing through the projection of visual features using an actionness-aware network. Our method's efficacy is demonstrated through experiments conducted on various tasks and baseline models. However, our model may be impacted adversely due to being dependent on the dataset labeling.

We discuss the complexities of the CrossTask dataset, particularly about challenging samples and their labeling. Figure 4.7 demonstrates instances of misclassified samples, despite utilizing d for post-processing. Notably, instructional videos often involve segments that are inherently challenging to classify due to their diverse styles, complicating the annotation process. Consequently, this could potentially lead to unreliable model inference and learning. For example, we observe an outlier scene in the video clip annotated with the "whisk mixture" action. The whisking motion is executed using a hand instead of a whisk, which our model misclassifies as a background. Similarly, the introductory section of the second video clip is labeled as a background, although it features the "cut lemon" action. Careful labeling and the removal of redundant scenes could improve model performance.

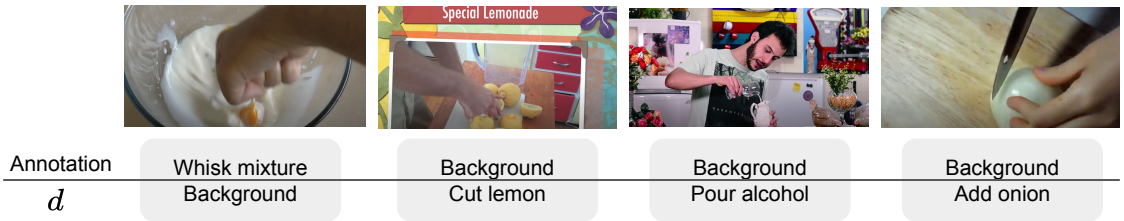

Figure 4.7 False predictions after post-processing with d. Coarse annotation of the dataset could be effective adversely as training of the d depends on the labels.

Our model predicts a background clip as "pour alcohol" despite the presence of the "pour water" action, as shown in the third column of Figure 4.7. This misclassification can be attributed to the absence of the "pour water" action step in the instruction set for the recipe "Make Irish Coffee". Hence, these action/background labels can mislead our model's training process. Moreover, the "add onion" and "cut onion" actions are annotated in separate tasks in the CrossTask. Assuming the cut and add actions happen consecutively, models tend to learn the similarity between the video clip and the "onion" word. It is highly possible to assign the "add onion" action to a video clip that includes onion cutting labeled as

background. To address this issue, the pre-defined action steps should consist of all main actions, and all video clips should be labeled according to them, not just the steps in the instruction set. Overall, our findings highlight the need for careful annotation and labeling strategies.

In the next chapter, instead of recreating or relabeling the data, we examine the action steps in the CrossTask datasets in detail and propose a data augmentation method to help with the mentioned issues.

5. IMPROVING ACTION/BACKGROUND DISCRIMINATION WITH IMAGES

The execution of actions portrayed in instructional videos on YouTube poses numerous challenges due to inconsistencies in tools, sequence or style of actions, surroundings, camera viewpoint, and materials employed. In Figure 5.1, we can see how many different ways the "add strawberries to cake" action is done in various videos. Compared to fundamental actions, instructional videos often feature more intricate and prolonged procedures. Ensuring consistency in this context presents a formidable task that may be unachievable. Our study utilizes ground truth labels for actions and backgrounds extracted from instructional YouTube video datasets to train a discrimination model. However, the effectiveness of this approach is reliant on the dependability of dataset labels, which is often problematic when dealing with complex data.

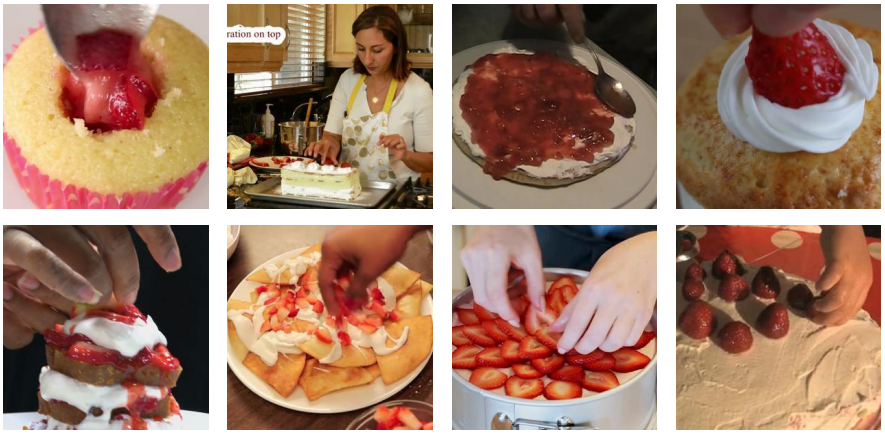

Figure 5.1 Different styles of executions for the "add strawberries to cake" action.

As discussed in Section 4.5., there are issues with the labeling of the CrossTask dataset from our research perspective. These problems include discrepancies in action step labels and their application to video labeling. For example, some actions are not labeled as such because they do not appear in the relevant task's action step list, while certain challenging scenes are incorrectly labeled as actions. Moreover, the action data is unbalanced across

different step labels, with some having few or no examples. This can mislead our simple action/background model, resulting in faulty post-processing of the probabilities.

To address this issue, we propose augmenting the training data with web action step images as done in [38] before, and we reduce the entire problem to an image-based format using the Contrastive Language-Image Pre-Training (CLIP) [39] model. We conduct all our experiments and analyses on a subset of the CrossTask dataset comprising only cooking tasks, which we refer to as **CrossTask-Cooking**. In this chapter, we exclude four primary tasks from CrossTask, namely "Jack Up Car, Change Tire, Add Oil to Car, Build Shelves".

Augmenting the training dataset with relevant image examples while preserving the discrimination between actions and backgrounds is a complex and demanding task. Several difficulties arise during the process, including (1) the insufficiency of action step labels leading to misleading data collection, (2) the usage of still images instead of videos creating a disparity between the original dataset and image data, (3) the bottleneck of the CLIP model in transferring the information of only shapes and colors instead of actions, which hinders the classification of many action step classes, and (4) the challenge of finding practical image examples that are action-specific since a recipe action could be employed in varying ways, as shown in Figure 5.1.

This chapter provides a comprehensive analysis of the CrossTask-Cooking dataset, with the aim of investigating the accuracy of its labeling and identifying any discrepancies in action step labels that may not be suitable for our research objectives. Subsequently, we present a detailed account of the data collection process from Google image search and the integration of the RecipeQA [3] image action dataset. We also provide the implementation details for augmentation and automatic image selection based on text-to-image similarities, along with the image baseline utilized for CrossTask-Cooking. These are followed by extensive presentations of our experiments and ensuing discussions. Our results reveal that the incorporation of images for all actions does not confer any significant advantage, but the use of reliable examples with appropriate scenes for specific actions can substantially enhance step recalls. Nevertheless, our intuition is that to harness the full potential of images

in defining complex actions, emphasis should be placed on hands, objects, their respective positions, and their interactions.

5.1. Comprehensive Analysis and Discussion of the CrossTask-Cooking Dataset

This section provides an overview of the annotation process for cooking actions in the CrossTask dataset and examines how challenging action labels can impact our study in various aspects. The original goal of the CrossTask dataset is to facilitate the sharing of information about similar actions across different tasks, which led to the use of weak supervision instead of detailed modeling for each separate step [33]. As a result, pre-defined action step labels are not highly detailed and descriptive, since the focus is on cross-task actions that have a covering meaning. For example, in [33], the "pour" action is learned using models that are trained for both "pour mixture" and "pour water". Here, the mixture is for either meringue or pancake. However, this approach can cause various challenges when evaluating current video-text models, since there is a lack of information about what the term "mixture" refers to. As video-text or image-text models are learned based on related narrations or captions, obtaining the similarity value between "pour mixture" and a scene may yield misleading results. A brief demonstration of this issue is provided in Section 5.4.1.. Furthermore, in order to collect images for action step labels, detailed descriptions are necessary. As such, our initial step is to identify which action step labels can be enhanced by incorporating additional nouns describing objects.

Following this, we discuss annotations conducted on video frames with accompanying action step labels that pose challenges. Such annotations bear the potential to misguide our model, and thus necessitate careful consideration. Lastly, we give the number of examples for each action step and the total number of backgrounds. Then, we mention the potential benefits of additional data collection for certain classes.

We limit our examination to the primary tasks and their associated action labels. We solely utilize videos from the **val-train** set, a subset of the original validation set. It is

used as the training set in our study. We exclude the examination of the original test and val-test/val-holdout sets presented in Section 3.1.1.. Our rationale is that addressing labeling issues would enhance the accuracy of experiments conducted on the CrossTask dataset in the existing literature.

5.1.1. Challenging Action Step Labels

During our analysis of the action labels and their associated video clips, we observe that some step labels are difficult to distinguish and associate with the corresponding action without knowing the related task. For instance, the action label "put mixture into bag" cannot retrieve related action images without information about what the "mixture" refers to. This situation brings an obstacle for image collection process and also affects the video-text or image-text similarities, which in turn affects the step recall results obtained on the CrossTask dataset, as demonstrated in Section 5.4.1.. To address this issue, we add extra words to some action labels without changing their meaning, and collect images from Google with the new labels. Figure 5.16 presents the list of action step labels for which images are collected. Modifications are made exclusively to those labels that have a markedly negative impact, while other labels are left unchanged unless additional information is deemed necessary but not compulsory.

Below, we present a comprehensive breakdown of each arduous action step label, consisting of the action step label and its corresponding task. Visual examples of frames are selected from the midsection of the action clips, to ensure the accurate representation.

- "add coffee" from "Make Latte": The analysis of annotated video frames (Figure 5.2) and the following second label "press coffee" indicates the necessity of specifying the form of coffee added, either as ground or whole bean. The possibility of confusion arises when adding brewed coffee. Due to varying styles, there is a need for special rearrangement; thus, the label for this step remains *unchanged*. This observation highlights the importance of clear and concise labeling in the preparation of coffee,

particularly when differentiating between the various forms in which coffee can be added.

Figure 5.2 Example frames for “add coffee”.

- “add ice” from “Make Lemonade”: The description of adding ice is overly broad. To enhance specificity, we employ the instance frames given in Figure 5.3 as a basis for relabeling the action as *“add ice cubes”*. This modification serves to refine and clarify the action’s definition.

Figure 5.3 Example frames for “add ice”.

- “add meat” from “Make Taco Salad”: In the context of this recipe, the term “adding meat” in the video frames (Figure 5.4) specifically pertains to the addition of ground meat. Despite the importance of providing a detailed description of the object, the label remains *unchanged* as it continues to facilitate the collection of images of ground meat and conveys the relevant information about adding meat, albeit without explicitly mentioning the term “ground”. Additionally, the ambiguity exists as to whether “adding meat” denotes the addition of cooked or raw meat. Nevertheless, this instance further highlights the challenging nature of recipes.

- “add taco” from “Make Taco Salad”: To ensure accurate image collection and contextual interpretation of the step label, we substitute the phrase “add taco” with

Figure 5.4 Example frames for “add meat”.

“add taco seasoning”. This modification is based on evidence from corresponding video frames (Figure 5.5) that demonstrate the actual action as adding taco seasoning rather than the taco itself. The inadequate specificity of “add taco” hinders both models and humans in comprehending the intended action without access to relevant contextual information.

Figure 5.5 Example frames for “add taco”.

- “add tortilla” from “Make Taco Salad”: Being a very specific action to the recipe taco salad, adding tortilla corresponds to adding tortilla chips, according to the example video frames in Figure 5.6. Similar to other modified step labels, even if it serves the main creation goal of the CrossTask dataset, it is an inadequate step label both for our image collection and baseline video-text or image-text models. Therefore, we change it to *“add tortilla chips”*.

Figure 5.6 Example frames for “add tortilla”.

- “add whipped cream” from “Make Irish Coffee”: Similarly, this action step label is lacking from the main object which the whipped cream would be adding on. As it is clearly comprehensible from the video frames (Figure 5.7), whipped cream is added on top of the coffee as a final step. Thus, we change the label to *“add whipped cream coffee”*.

Figure 5.7 Example frames for “add whipped cream”.

- “close lid” and “open lid” from “Grill Steak”: The corresponding frames (Figure 5.8) demonstrate closing or opening a grill’s lid for this step label. That being so, we replace them with *“close grill lid”* and *“open grill lid”*.

Figure 5.8 Example frames for “close lid” and “open lid”.

- “mix ingredients” from “Make Banana Ice Cream”: Based on the recipe and the video frames (Figure 5.9) annotated with the label “mix ingredients”, it can be inferred that all the added ingredients are mixed through a blender rather then other mixing tools. Therefore, it seems it is a special action step for this recipe, and adding an extra object is crucial. Despite the challenge in identifying a distinct label for this process, *“mix ingredients blender”* is utilized instead of the aforementioned label.

- “pour egg” from “Make Meringue, Make French Toast, Make Strawberry Cake, Make Pancakes”: The demonstration of this action is multifaceted, with instances ranging

Figure 5.9 Example frames for “mix ingredients”.

from a simple pouring of the egg from a dish to the cracking of the egg into a mixture or solely pouring the egg yolk (Figure 5.10). The presence of this step across diverse recipes underscores the challenge in comprehending specific actions, owing to their disparate contexts and applications. We leave the step label *unchanged* since there is a requirement for multiple step labels and relabeling of the data.

Figure 5.10 Example frames for “pour egg”.

- “pour mixture into cup, put mixture into bag, remove bread from pan, spread mixture” from “Make Jello Shots, Make Meringue, Make French Toast, Make Meringue” respectively: These step labels are changed as “pour *jello* mixture into cup, put *meringue* mixture into bag, remove *french toast* from pan, spread *meringue* mixture” in order to describe what the mixture and the bread refer to. The alterations are imperative due to the overly broad definition of mentioned action step labels.

- “put dough into form” from Make Strawberry Cake: Upon analysis of the video samples in Figure 5.11, it has been established that the act in question may be more aptly described as the process of transferring dough or cake mixture into a cake mold as opposed to the present interpretation of kneading the dough. Nevertheless, the label remains *unchanged* as it requires rearrangement of the labeling.

Figure 5.11 Example frames for “put dough into form”.

- “pour water” from “Make Jello Shots, Make a Latte, Pickle Cucumber, Make Lemonade, Make Fish Curry”: Similar to “pour egg”, this action has varying styles per task since the goal for pouring water differs, such as pouring into coffee, a jar, a glass, or a pan (Figure 5.12). This step label can be multiplied with additional phrases defining where the water will pour. Since it needs a relabeling process to fix the ambiguity and to be helpful for our problem, the label stays *unchanged*.

Figure 5.12 Example frames for “pour water”.

- “stir, stir mixture, whisk mixture”: These action step labels define stirring and whisking actions, which are most common for many tasks. Not having them as more descriptive actions brings challenges similar to mentioned labels. However, in order to make them action specific, we need corresponding objects, such as stirring the salad or whisking the cake mixture. Since the additional nouns will differ from task to task, we leave these labels as *unchanged*.

5.1.2. Challenging Annotations

In Section 4.5., we present certain scenarios where actions lack annotation with respect to action step labels, due to their exclusion from the designated task’s action step list. Such an

approach towards labeling misguides our research, as it employs the dataset labels to learn the action/background discrimination model. Our examination of annotated video frames further reveals additional challenges that can significantly affect our study and related literature. Consequently, in the subsequent items, we provide a detailed breakdown of the annotation problems prevalent in the val-train set video frames.

- Annotating the narrators: Within some videos, the corresponding action step label is assigned to the scenes where the narrator only speaks. The validity or invalidity of this practice is questionable, as the actions in instructional videos on YouTube typically involve a demonstration that spans the entirety of the action. Although in the CrossTask dataset the actions are annotated as chunks, refining and adjusting the action labeling process may enhance the evaluation. Furthermore, our study is subject to potential impact, as a background scene is utilized as a positive sample. Figure 5.13 displays example video clips for various actions where the narrator appears before, in the middle, or after the action and is annotated with the action step label.

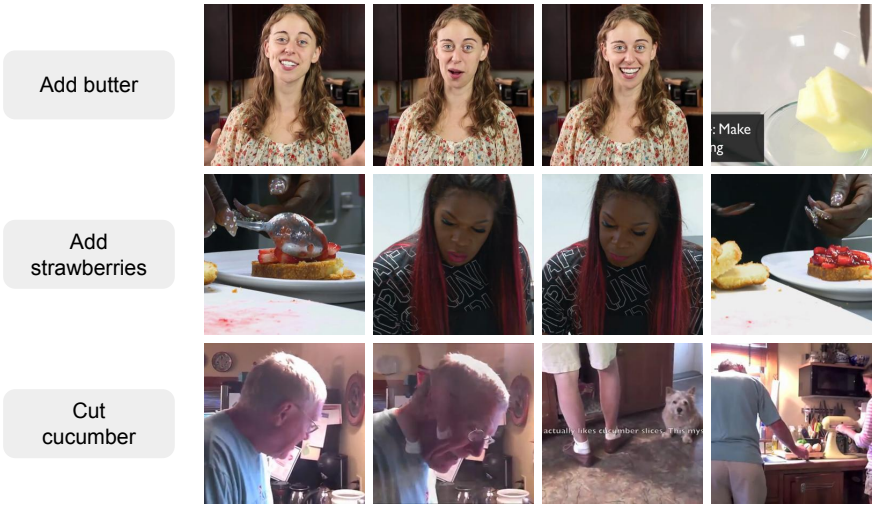

Figure 5.13 Example video clips for specific actions where the narrators are annotated with action step labels.

- Faulty annotation: Furthermore, it is imperative to consider the accuracy of labeled scenes in addition to the labeling of narrators. It has been observed that certain labeled scenes lack relevance to the actions depicted and lack any action items,

indicating faulty labeling. The examples provided in Figure 5.14 exemplify this issue, emphasizing the need for a meticulous double-check during the dataset labeling process. Such attention to detail will ensure the reliability and validity of the dataset. Even though these labels do not affect positive data and thus our action/background model, as we explain in the upcoming experimental sections, they affect our image selection process and, of course, the evaluation in the general tasks.

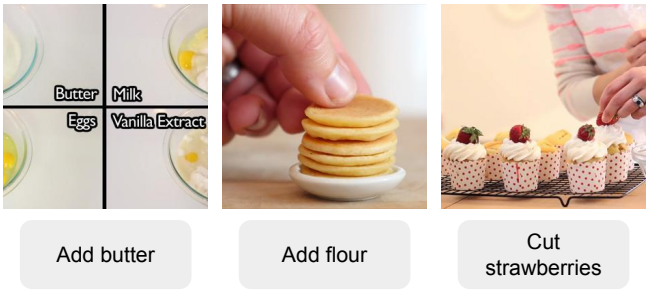

Figure 5.14 Examples of faulty annotations.

- The discrepancy between the recipe applied in the video and the action step label: As cooking videos allow for flexibility in ingredient choice and preparation methods, viewers may adapt a recipe in various ways to suit their preferences. For instance, in the video tutorial "Make Kimchi Fried Rice", alternative ingredients such as chicken or shrimp are used in place of ham, despite the task only requiring the addition of ham. The corresponding frames are still labeled as "add ham". Similarly, a narrator adds chicken instead of meat to a taco salad, and this action is labeled as "add meat". Another example involves the injection of jello mixture into chocolate instead of pouring it into a cup, while the corresponding action is labeled as "pour mixture into cup". Figure 5.15 presents these examples. Although these instances may be outliers, improving the labeling process by utilizing more detailed phrases and employing greater care in labeling would bolster the reliability of the dataset.

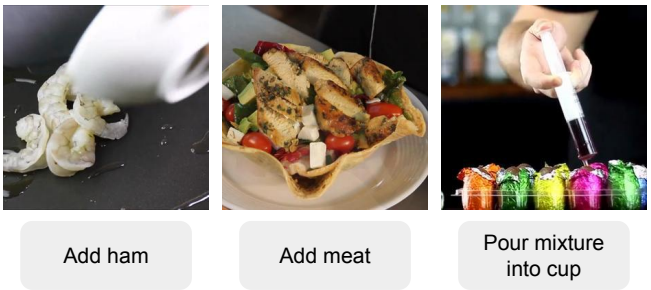

Figure 5.15 Examples of discrepancies between the action in the video and the annotation.

5.1.3. The Number of Positive and Background Data

This section presents the sample distribution for each action step comprising the positive dataset, along with the total number of backgrounds employed in the study. The variance in sample count across action steps is visualized in Figure 5.16. The aggregate number of positive instances and backgrounds utilized in the study is 11794 and 31831, respectively. Note that this set corresponds to the cooking subset of the val-train split of the original validation data. Unlike val-train in Section 4., here we have 168 videos since four non-cooking tasks are excluded.

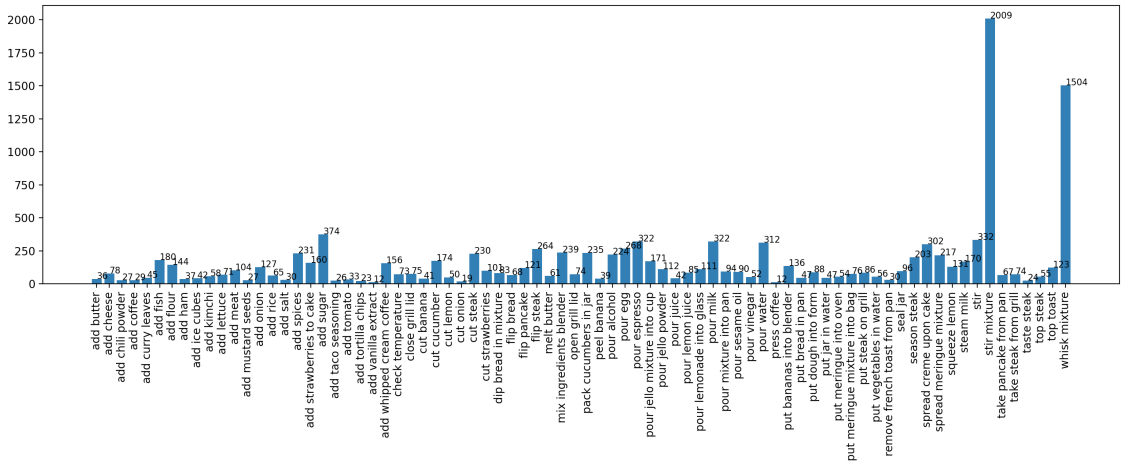

Figure 5.16 The number of samples for each action step in the CrossTask-Cooking val-train.

The analysis of Figure 5.16 reveals that the number of examples varies across the action steps due to the duration and frequency of occurrence of each action. For instance, the action step “stir mixture” generally requires more time to execute and has more frames compared

to the other steps. Moreover, the action step “add sugar” is annotated with a greater number of distinct video clips. On the other hand, in our val-train split, the action step “cut onion” is represented by only one video clip, consisting of 19 frames. Likewise, “press coffee” has only 12 examples from two different videos. These circumstances render recognition of certain actions as positive action classes more challenging, as learning is based solely on inadequate examples. In this chapter, we aim to augment each action step data in quantity and quality with practical instances to improve our model learning.

5.2. Datasets

In this section, we present the image datasets employed to examine the concept of enhancing instructional video datasets with action images. We initially collect images from Google search using action step labels. Then, we select appropriate data for each action by hand and we utilize the pre-defined action step labels to identify the most similar action images. Furthermore, we experiment with RecipeQA images. We gather images through leveraging the image-text similarity provided by the CLIP model.

5.2.1. Google Images Collected with Modified Labels

We obtain images for each modified action step label in the primary tasks of CrossTask-Cooking, as presented in Figure 5.16. To collect these images, we utilize the Google image search API through the open-source GitHub repository[4]. In order to expand the dataset, we augment the step labels with the terms “cooking” or “recipe” and search for 300 images for each category, resulting in a subset of photos that are more relevant to the cooking domain. We then merge the two categories into a single class for the purposes of our experiments. After removing duplicates, we obtain approximately 500 images for each action step label. Notably, we exclude the action step “move steak on grill” from our queries, as this step is not present in the original training set val-train. The overview of the total number of collected images is given in Figure 5.17.

[4]https://github.com/ohyicong/Google-Image-Scraper

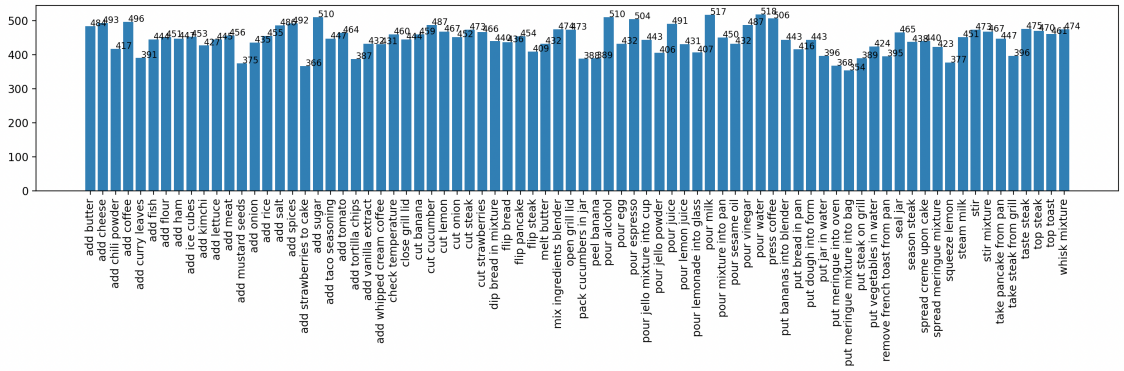

Figure 5.17 The total number of Google images collected for each action step label.

We propose to investigate the efficacy of image augmentation method in improving the performance of action recognition using web images. Along with the proper instances, web images are known to contain noisy and complex information, posing a challenge to access useful data. It is possible to collect an image including just the objects/scene instead of the action itself or just an illustration of the action without the environment as a special shot rather than a natural action image. Moreover, irrelevant examples may arise due to the presence of non-task/recipe specific action step labels. An example of collected images from the top rows of Google search is shown in Figure 5.18. As such, further investigation is necessary to determine the suitability of the image augmentation method. To this end, we construct a clean hand-selected subset comprising the most relevant images from each action class (Section 5.4.2.). Additionally, we utilize text-to-image similarities between pre-defined action step labels and images for automatic selection (Section 5.4.3.).

5.2.2. RecipeQA

As a supplementary dataset for image analysis, the training set images of RecipeQA [3] are utilized. The images in RecipeQA are comprised of instructional depictions for particular recipes, which have been collected from the Instructables website. This platform offers a vast array of recipe action steps accompanied by detailed descriptions and corresponding images. The training set comprises 109024 images, which were gathered for various question answering (QA) tasks. The example images for the recipe “3 minutes vegan cake” in the

Figure 5.18 Example of top Google images for “squeeze lemon”.

RecipeQA dataset are given in Figure 5.19. Similar to web videos or Google images, the dataset is noisy. Moreover, most of the scenes include the result of the action rather than the action itself. There are no pre-defined action labels for each instruction. Therefore, we exclusively extract the most pertinent images by relying solely on the pre-defined action step labels of CrossTask-Cooking. We utilize the text and image encoding similarities obtained from the CLIP model. The elaborate account of our experiments is presented in Section 5.4.3..

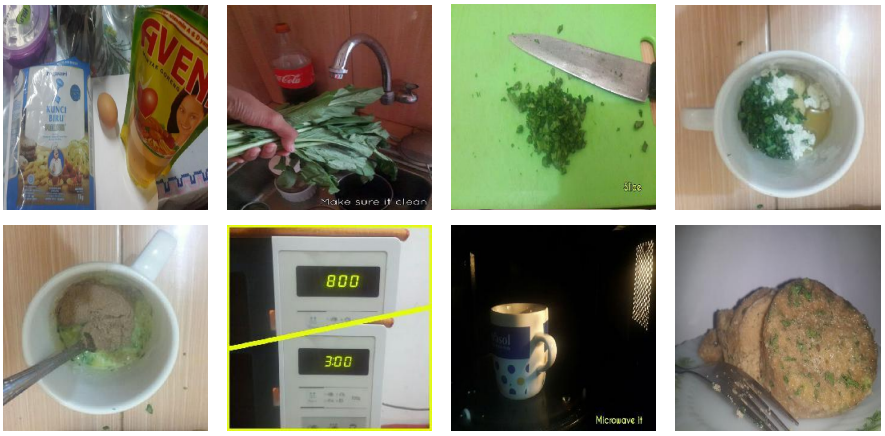

Figure 5.19 Example instructional step images for “3 minutes vegan cake” recipe in the RecipeQA [3].

5.3. Implementation Details

In this chapter, we use Contrastive Language-Image Pre-Training (CLIP) [39] model to obtain features for video frames, images, and action step labels. CLIP is trained with 400 million image-text pairs collected from the web where the text includes at least one word from the query list covering 500000 items. They use Vision Transformer (ViT) [105] along with different architectures for the image encoder and Transformer [106] for the text encoder. In this work, we utilize ViT-B/32 as it is done in [70]. After having L2 normalized features from a shared encoding space in the dimension of $1\text{x}512$ both for an image and the text, the cosine similarities are calculated. Softmax is applied to obtain the probability distributions. We use probabilities for text-to-image retrieval as in [39], but we use cosine similarities to regenerate the CrossTask baseline as the action step assignment method from [1, 2, 33] use those.

5.4. Experiments

5.4.1. CLIP Baseline

For the CLIP-based CrossTask baseline, we choose one random frame from each one-second clip. After having the features of frames and base action step labels, we obtain cosine similarities. We follow exactly the same approach in Section 4. and use the same prediction and evaluation method with the same original test set. The average step recall is calculated as 16.9. While it has been previously reported that CLIP can lead to reduced accuracy for video data due to a lack of temporal information [39], we propose that an additional issue is the scarcity of action-containing examples in the CLIP training data, which we explore further in Section 5.4.3.. Furthermore, after excluding the non-cooking tasks from the dataset, we calculate the baseline for CrossTask-Cooking as 18.73. In the present study, we refine the actionness score computation by training d and d_v solely on video frames obtained from video clips in the val-train set. The updated results are reported in Table 5.1, where d_v denotes the actionness model trained using exclusively visual features, and d denotes the model

jointly trained with auxiliary features. The baseline is post-processed using the actionness model scores, as proposed in Section 4. and our prior research [4]. To enhance the positive set of visual features, we incorporate supportive images in this chapter. Therefore, we consider the outcomes of d_v as our starting point and aim to enhance it further by utilizing additional data.

	AP	SR
Baseline	-	18.73
Baseline - Modified	-	19.65
d_v [4]	49.77	19.85
d [4]	50.03	20.16

Table 5.1 Average precision (AP) of action/background discrimination and average step recall (SR) results for CrossTask-Cooking dataset with CLIP image-based features.

As previously discussed, the CrossTask-Cooking baseline can be enhanced by adjusting action step labels prior to step assignment. This improvement is necessary as some labels lack specific information and are overly broad without proper context. Leveraging video-text or image-text models such as CLIP that depend on text-to-image similarities, the addition of descriptive language to these labels has the potential to improve the baseline. Our hypothesis regarding the significance of modified action step labels is confirmed by the results presented in Table 5.1 (Baseline - Modified). The modified labels are used in the prediction process without any post-processing. Despite the promising results, the current performance still falls short of that achieved through post-processing with actionness (d, d_v), demonstrating the continued relevance of the actionness score.

5.4.2. Hand-selected Subset of Google Images

To evaluate the validity of the proposed image augmentation technique, we construct a subset of Google images that are highly representative of the targeted action. The selection criteria is based on the similarity of the images to the action frames present in the CrossTask val-train set. Additionally, we emphasize the selection of images that portray the action more effectively than the corresponding frame or those with unique features. For challenging

task-specific actions, including “stir mixture, stir, seal jar, put vegetables in water, put dough into form, pour juice, dip bread in mixture, flip bread, pour egg, press coffee”, we select contextually similar images since obtaining related images proved challenging. Consequently, despite the objective of constructing a purified subset, a few of the collected images may still exhibit undesirable outcomes, instead of improving the discrimination between action and background. Figure 5.20 illustrates the number of total examples after augmenting val-train video clips with hand-selected Google images. The presented findings, reveal noteworthy insights regarding the quantity of selected examples. While a satisfactory number of samples are attainable for certain action steps, a considerable number of samples are noticeably absent for others. In the following, we present several instances of problematic and satisfactory images. Then, we report the outcomes of enlarging the training dataset by incorporating all selected images, as well as solely incorporating a subset of good quality ones. A detailed analysis of the generated outputs is then provided.

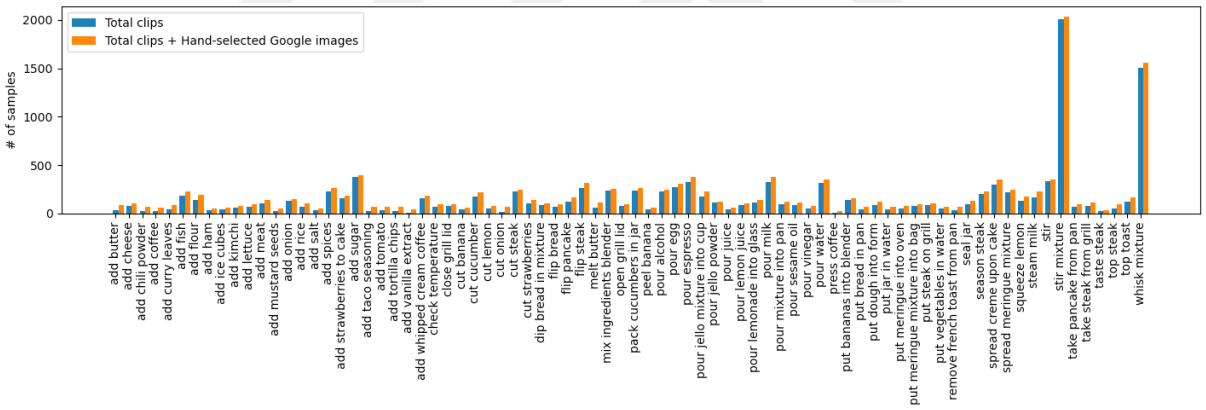

Figure 5.20 The number of total samples in the val-train set before and after augmenting the number of clips with the hand-selected subset of Google Images.

5.4.2.1. Quality of the Examples: Figure 5.21 displays a set of images that are chosen to represent the related action step. However, these images are insufficient for discriminating between actions, as they did not depict the relevant action or any action at all. The reason is that searching for some specific action step labels is challenging such as “press coffee, add spices, add salt, dip bread in mixture”. For instance, searching for “press coffee” yields images of french presses, while searching for “add spices, add salt” yields images related to

general recipes, rather than those specific to pickle recipes. Similarly, when searching for "add kimchi, add tortilla chips," only images of the resulting products are available, rather than those that capture the action itself. Such difficulties are expected, as it is challenging to find images that depict specific actions without a detailed description. On the other hand, augmenting action step labels with recipe/task-specific descriptions can lead to inadequate data, as it reduces the possibility of finding relevant images. This situation is not desirable as well. Therefore, we select images that are contextually similar to action step frames such as scenes featuring tortilla chips, kimchi dishes or adding spice into pot. Despite having a few good examples, we often get insufficient data for these action classes, as shown in Figure 5.20.

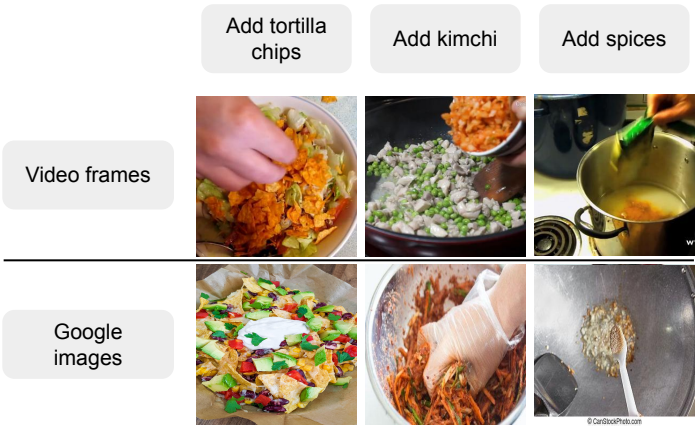

Figure 5.21 Original video frames and difficult examples selected from Google Images for action steps.

In Figure 5.22, we present good examples that represent the related action step and are similar to video frames. As such, the inclusion of these images has the potential to bolster the discrimination of corresponding actions by increasing both the quantity and quality of the available samples. It is worth noting, however, that the acquisition of useful images is limited to a select few action step classes. In the following section, we present our findings on the impact of integrating both effective and ineffective examples on discrimination, and subsequently, post-processing.

Figure 5.22 Representative Google images selected for action steps.

5.4.2.2. Results: We report on the inclusion of 2439 manually selected images as positive samples in addition to video frames and its effect on action/background discrimination (AP) and average step recall (SR) given in Table 5.1-d_v. We use the same positive and background video frame examples for the analysis. Our goal is to enhance the aforementioned performance metrics. We add samples to val-train and train the binary discrimination model d_v as in Section 4.1.1. - Equation 5. We post-process the similarity scores of the original test set with the actionness scores of d_v as in Section 4.1.2. - Equation 7. Then, we rerun the dynamic prediction (Section 3.1.3. - Equation 1) as it is mentioned in Section 4.4.1.. Note that in the following experiments, the combination parameter α is decided as 0.9 by tuning on val-holdout. We conduct a series of experiments. The first experiment incorporates all image samples from 78 action step classes, while the second experiment involves a narrowed focus on only 4 action step classes, which exhibit high quality. Lastly, the third experiment involves the inclusion of images from the "steam milk" category and a thorough analysis of the findings. We present the results in Table 5.2. Here, "random" represents the set created by choosing 20 random images from each action step class.

From Table 5.2, it can be inferred that not adding all selected images but the high-quality ones improves the results. The main problem is majority of the classes have still images instead of the actions and forcing the binary model to take them as positives can increase false positive predictions on backgrounds even more. As we mentioned before, since the

Image Set	AP	SR	# of images
Baseline (d_v [4])	49.77	19.85	-
Random	47.44	19.32	1560
All	47.56	19.38	2439
Best 4	50.20	20.01	190
Steam Milk	50.66	20.22	61

Table 5.2 Average precision (AP) of action/background discrimination and average step recall (SR) results when val-train is augmented with different hand-selected Google image sets.

instructional action steps are complex and prolonged, finding proper images is a challenging but interesting problem.

We further examine the step based results in Figure 5.23 to infer which classes are poorly affected and vice-versa. We separate 4 of the best improved classes which also contain high quality Google images. This subset contains "cut onion, pour mixture into pan, spread creme upon cake, steam milk" action step classes. Further experimentation revealed that adding more categories does not enhance performance but decreases it since the samples have a harmful impact across other action step classes, as can be seen from the results of adding "All" images. Thus, we limit the high-quality subset to only these 4 classes.

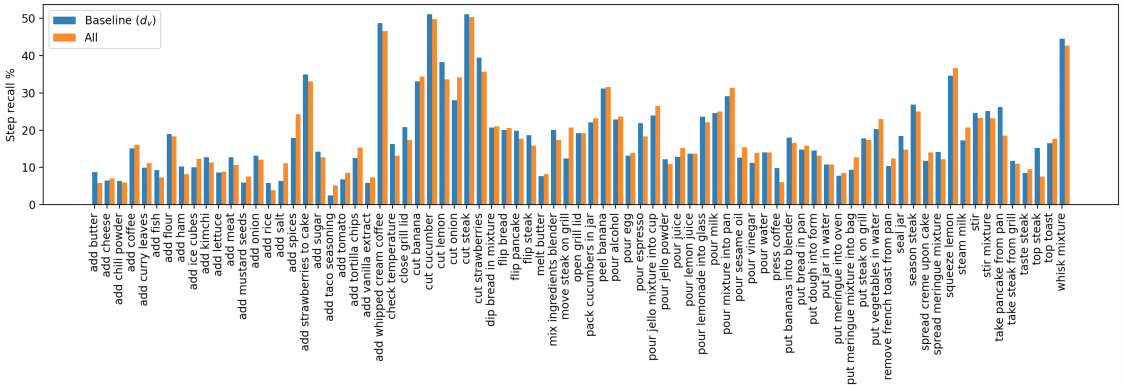

Figure 5.23 Comparison of step recall results for "Baseline (d_v [4])" trained with val-train and "All" trained with val-train + hand-selected Google images.

In Table 5.3, we present the step based results for image augmentation with the "Best 4" subset. The present findings indicate that the incorporation of images can effectively enhance the classification performance for specific actions, particularly in the case of "cut onion",

where the baseline dataset comprises a limited number of instances. Such an improvement highlights the significance of the quality of samples utilized in the training process. Finally, we see that high-quality "steam milk" images contribute little to the action discrimination of "steam milk", but they are the images that increase the overall average step recall the most when they are used alone (Table 5.2). Therefore, we investigate this further.

Image Set	Cut Onion	Pour Mixture into Pan	Spread Creme upon Cake	Steam Milk
Baseline (d_v [4])	28.07	29.12	11.73	17.29
Best 4	34.36	30.00	13.86	18.93

Table 5.3 Average step recall results for 4 action step classes when image augmentation with the hand-selected "Best 4" is applied.

We explore the impact of adding "steam milk" images to the positive set on the step recall of other action steps. Specifically, we analyze the step-based results obtained from this augmentation to gain insights into the effect of qualified images on the discrimination of different action classes. The results presented in Table 5.4 demonstrate a significant improvement in the step recall of four other action steps. To illustrate this finding, we provide video frames depicting these actions in Figure 5.24, which show scenes, objects, structures, and colors similar to those of a "steam milk" image (Figure 5.22), even though the actions themselves are unrelated. For example, in the "add milk" scenes, the liquid and container can be visually similar, and it is difficult to distinguish objects and colors in "add sugar" or "put dough into form" scenes. Notably, actions involving hands, such as "pack cucumbers in jar", are also found to be improved, likely due to the similarity in scene structure when holding objects. While these findings are intriguing, further investigation is needed to better understand their implications. Establishing a standardized criterion for identifying the specific image exemplars that impact distinct action steps remains challenging, given the limitations of the CLIP bottleneck. Additionally, this experiment highlights a potential limitation of the CLIP model's ability to understand actions from images, as discussed in Section 5.4.3.. We demonstrate that the primary training data for CLIP comprises image-text

pairs containing non-action contexts, which results in the model's difficulty in encoding actions.

Image Set	Pour Milk	Add Sugar	Pack Cucumbers in Jar	Put Dough into Form
Baseline (d_v [4])	24.63	14.27	22.15	14.57
Steam Milk	28.50	19.43	30.02	20.27

Table 5.4 Average step recall results of some improved action step classes when image augmentation with the hand-selected "Steam milk" images is applied.

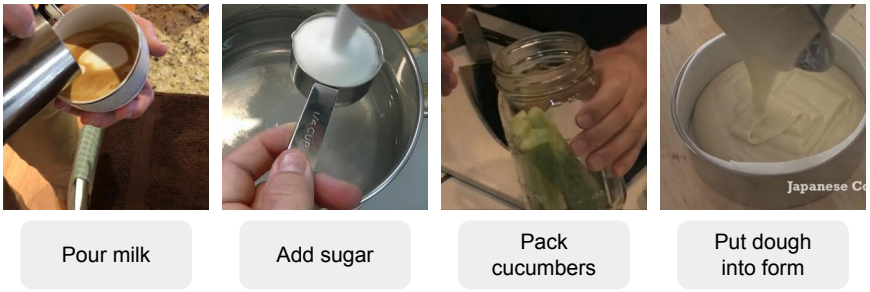

Figure 5.24 Example video frames for specific action step classes where hand-selected "steam milk" images contribute to their discrimination.

Furthermore, we investigate the efficacy of using only hand-selected Google images in training set while excluding all video clips. Our findings demonstrate that (Table 5.5) learning a discrimination model solely from images is not feasible due to the natural domain gap between video clips and images, as previously mentioned in Section 5.2.1. and discussed in [94]. Then, we employ these hand-selected images to identify relevant video clips that depict the action step by assuming the chosen images' representation of the action. Using a more confident set, the primary objective is to eliminate challenging video frames described in Section 5.1.2.. To this end, we utilize hand-selected images as seeds for the related action step and calculate the cosine similarities between video clips. We select video clips with an average similarity value of at least 0.65 with all seeds. We determine this threshold empirically and give the number of selected clips in Figure 5.25. However, as shown in Table 5.5, our results indicate that this approach does not improve but decreases the step recall. As described in Section 5.4.2.1., the challenges associated with selecting images highlight the complexities of identifying suitable images to represent

intricate actions. Nonetheless, it is also essential to consider the CLIP bottleneck as an effective issue.

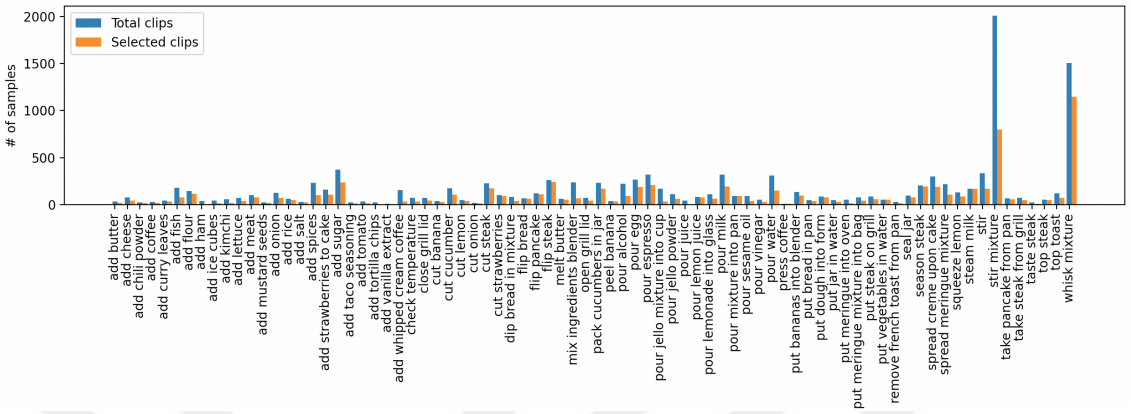

Figure 5.25 The number of clips chosen by using hand-selected Google images as seeds.

Training set	AP	SR	# of positive data
Baseline (d_v [4])	49.77	19.85	11794
Hand-selected Google images	28.37	16.67	2439
Selected clips with image seeds	49.14	19.54	7379

Table 5.5 Results for additional training set variants.

5.4.3. Auto-selected Subsets of RecipeQA and Google Images

We demonstrate the use of the CLIP model's text-image similarity capability to identify relevant images for action step categories. Instead of manually selecting a subset, we rely on the cosine similarities between the text encoding of action step labels and the image encoding to obtain the most relevant images. We apply softmax to the cosine similarity values of each image to obtain logits, as previously done in [39]. Using a threshold of 0.95, we eliminate images with low scores. This threshold is determined through empirical examination of the selected images. We apply this process to both the collected Google images and the RecipeQA training set, augmenting the positive video set with the auto-selected images. Note that we use modified action step labels, and our objective is to improve the "All" result in Table 5.2 by filtering out bad examples. However, our results, as shown in Table 5.6, indicate that automatically determining images for improving action/background

discrimination remains a challenging open research problem. We attribute this difficulty to the CLIP bottleneck, which may have a significant impact on the selection of good examples.

Image Set	AP	SR	# of images
Baseline (d_v [4])	49.77	19.85	-
RecipeQA	47.33	19.57	792
Google	47.31	19.66	1973

Table 5.6 Average precision (AP) of action/background discrimination and average step recall (SR) results when val-train is augmented with auto-selected image sets collected according to text-to-image similarities.

We give examples of selected images in Figure 5.26. It can be seen that the images achieved the highest similarity scores for their corresponding action step labels are dominated by static objects and text rather than depicting the relevant actions. These findings highlight the need to train the CLIP model with verb-centric image pairs to address the identified limitations.

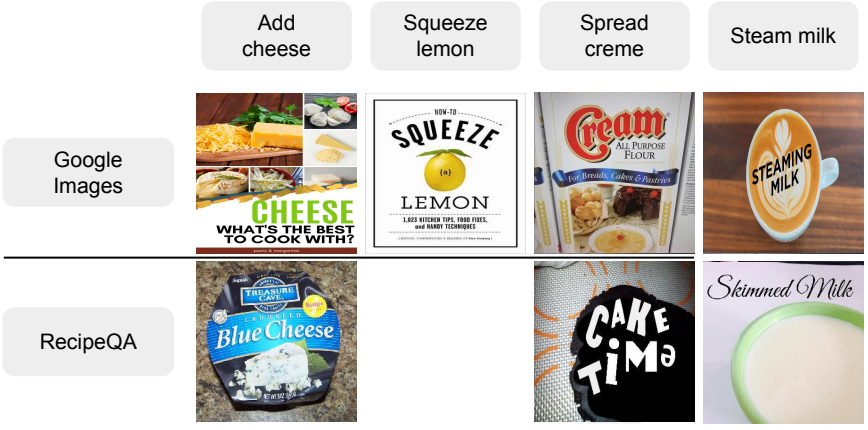

Figure 5.26 Example images selected with text-to-image similarity values of CLIP.

5.5. Discussion

The present study comprises experimental investigations into the efficacy of augmenting video samples with images as a means of enhancing the action/background discrimination model. The findings indicate that this approach is a complex process and, on balance, does not provide a superior solution compared to the joint training of auxiliary features as outlined

in Section 4.. However, the study identifies a promising research field that merits further exploration. In the following, we discuss the possible problems and how the images can be leveraged in the future. The foremost challenge pertains to the CLIP bottleneck, which arises from the model's training data being designed for recognizing visual concepts. Fine-tuning the model with action step label-image pairs can bolster the action recognition capability of CLIP. The second issue is related to the fact that the utilized images do not capture the human background context in which actions occur, unlike in YouTube instructional videos where the demonstrator is depicted performing the action.

Additionally, it is essential to note that images often capture only a single state, such as a fully prepared dish or an individual ingredient. Conversely, videos capture a temporal sequence of actions, such as the process of adding an ingredient to a pan. The use of isolated images to represent a single action phase, such as "adding tomatoes", may limit the learning potential of the model to a single step. This issue can be resolved by segmenting an action into multiple phases consisting of initial, intermediate, and resulting states and collecting relevant images for each stage.

The objective of this study is to enhance action localization by refining the action/background discrimination model through the use of high-quality training data. The current labeling methodology employed by CrossTask is inadequate due to its coarse nature. Given the complexity of actions, which entail multi-stage processes, object manipulation, hand movements, scene interaction, and various phases, single action images cannot suffice for improving data quality. Consequently, datasets with a similar structure to the existing data could serve as better sources of information. Moreover, objects often signify action steps, as in "put steak on grill", where the grill and the steak define the scene. Thus, the importance of object interactions should be noticed while integrating images and videos.

6. CONCLUSION

In this thesis, we attack the problem of understanding fine-grained complex actions in instructional web videos. To this end, we focus on improving the action localization task evaluated on annotated datasets with pre-defined action step labels. We examine the robustness of utilized baseline video-text models that provide similarity scores for assigning action step labels to video clips. Our investigations show challenging background clip scores affect action localization accuracy. Motivated by previous work, we aim to mitigate their effect by modifying similarity scores as a post-processing step before doing assignments. The goal is to learn the *actionness* of video clips and decrease or increase similarity scores accordingly.

To address this, we first suggest reinforcing the recognition of backgrounds and actions since visual encodings present inadequate discrimination according to outlier background scores. We introduce a novel representation for video clips containing cues from their similarity values for action step labels. We hypothesize that while action clips obtain confident scores for specific action steps, background clip scores will present high aleatoric uncertainty through action step labels. We propose handling the similarity scores of a video clip by converting them into a probability distribution. Then, we extract four different components based on the highest value, the entropy of the vector, and neighbors. In Section 3., we illustrate that these components help the discrimination of action clips from backgrounds.

After showing the contribution of similarity cues in action/background discrimination, we explore using them for learning an actionness model in Section 4.. We extend the idea of similarity cues and define an auxiliary representation directly from the probabilities of the video clip and its neighbors. We suggest learning a binary discrimination model by utilizing a small annotated set. The model is trained jointly with visual and auxiliary representations of video clips using shared weights. The goal is to project visual features into a more action/background discrimination aware space so that the output probability can depict the actionness of a given video clip. We conduct action localization and action

segmentation experiments, in addition to ablation studies, on CrossTask [33] and COIN [36] datasets, respectively. We show improvement over the previous work and present a new post-processing approach for the action segmentation task.

We argue that not the number of data but the quality of the data affects the action/background discrimination. Since we rely on action and background labels, challenging annotations can affect our method. Therefore, in Section 5., we investigate augmenting the subset of the CrossTask [33] with action images collected from the web for an improved action/background discrimination model. We first analyze the challenges we face due to the structure of the CrossTask dataset from our study's perspective. Then, we mention the difficulties of collecting proper image web data for fine-grained instructional action steps. We leverage the image-text foundation model CLIP [39] to encode video frames and images. We present primarily results with a hand-selected subset of web images. The outcomes suggest a promising future direction when the selected samples are proper for the related action. However, as we discuss in Section 5.5., finding good examples and the model we utilize pose a challenge in the way.

6.1. Limitations and Future Directions

Recognizing fine-grained instructional actions from web videos is a complex problem from the computer vision aspect. It is affected by various challenges:

- Actions appear in different scenes due to variations in ingredients, tools, and environment.
- People demonstrate actions and instructions in different ways. One can pour the egg by cracking it directly, while another can pour it from a cup. Moreover, people can improvise and tutor tasks in their style.
- Camera viewpoints vary due to the shooting styles of the videos.
- Automatic transcriptions can be misleading since they depend on the person's narrative style.

- Presence of many background scenes increases the complexity of the problem.

According to these difficulties we observed during our study, digging into foundations should be the next step. Regarding baseline model robustness, one crucial thing to handle in the future could be leveraging multiple modalities such as audio, object tokens/interactions, and hand positions as additional supervision. Moreover, benchmark datasets should be labeled more carefully as annotation of fine-grained actions can be subjective, and challenging labeling can mislead the evaluation of action localization.

In this study, we use pre-trained embedding models for our baseline scores, representations, and video-text similarity values. These models can be replaced with more updated ClipBERT [42] or CLIP4Clip [19] to get enhanced results. However, some state-of-the-art works present more specialized networks for understanding instructional videos [63, 107, 108]. In [63], they propose to use wikiHow instructions to match them with automatic transcriptions and then to learn joint embeddings between related video snippets and wikiHow instructions. Furthermore, in [108], they also suggest encoding the relation between wikiHow instruction steps and refining features from a video embedding model with more procedure-aware information. Thus, taking these models as the base and leveraging multi-modalities simultaneously would be a helpful starting point for the problem.

We propose to handle the adverse effect of misleading background scores by post-processing them. Although we show improvement by modifying scores with our actionness aware method, there are many things to consider in the future. First of all, auxiliary representation could be inadequate for defining the edge video clips that start or end an action. We do not handle this situation, and using neighbor properties can give a misleading description for them. Moreover, we have a minor enhancement on the COIN dataset for the action segmentation task. While changing all scores, we may also lose true positives due to bad actionness scores. The reason is that in the COIN dataset, we also modify background scores and calculate accuracy for detecting them. However, without having a near-perfect actionness model, post-processing has drawbacks too. Therefore, detecting outliers and only modifying their scores could be one step.

To address this issue, we suggested learning an improved action/discrimination model next. We observe that our training set labels can have an impact. Thus, we augment the dataset with images and show a promising direction. According to challenges mentioned in Section 5.5., image-to-video adaptation models that transfer the knowledge between two domains can be explored and leveraged for improved representations. As a result, action/background discrimination could be enhanced.

Finally, in this field, a user study is needed. The next question would be how much we can learn instructions as humans from given YouTube videos by recognizing the fine-grained actions, before asking models to understand them. In such a challenging environment, a human upper bound for the benchmark datasets would provide a helpful direction for the interpretation of developed models.

REFERENCES

[1] Dimitri Zhukov, Jean-Baptiste Alayrac, Ivan Laptev, and Josef Sivic. Learning actionness via long-range temporal order verification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 470–487. Springer, **2020**.

[2] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889. **2020**.

[3] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368. **2018**.

[4] Ozge Yalcinkaya Simsek, Olga Russakovsky, and Pinar Duygulu. Learning actionness from action/background discrimination. *Signal, Image and Video Processing*, pages 1–8, **2022**.

[5] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, **2004**.

[6] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, **2008**.

[7] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition.

In *2011 International conference on computer vision*, pages 2556–2563. IEEE, **2011**.

[8] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, **2012**.

[9] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970. **2015**.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308. **2017**.

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732. **2014**.

[12] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, **2014**.

[13] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634. **2015**.

[14] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In

Proceedings of the IEEE international conference on computer vision, pages 4489–4497. **2015**.

[15] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702. **2015**.

[16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, **2016**.

[17] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213. **2020**.

[18] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846. **2021**.

[19] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, **2022**.

[20] De-An Huang, Joseph J Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2183–2192. **2017**.

[21] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding" it": Weakly-supervised reference-aware visual

grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5948–5957. **2018**.

[22] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, pages 334–350. Springer, **2020**.

[23] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172. **2020**.

[24] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15611–15620. **2021**.

[25] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, **2022**.

[26] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13956–13966. **2022**.

[27] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119:346–373, **2016**.

[28] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *2016 IEEE Winter*

Conference on Applications of Computer Vision (WACV), pages 1–8. IEEE, **2016**.

[29] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, **2016**.

[30] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*. **2018**.

[31] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583. **2016**.

[32] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. **2018**.

[33] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545. **2019**.

[34] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640. **2019**.

[35] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473. **2019**.

[36] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216. **2019**.

[37] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. *Advances in neural information processing systems*, 33:22605–22618, **2020**.

[38] Shugao Ma, Sarah Adel Bargal, Jianming Zhang, Leonid Sigal, and Stan Sclaroff. Do less and achieve more: Training cnns for action recognition utilizing action images from the web. *Pattern Recognition*, 68:334–345, **2017**.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, **2021**.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, **2018**.

[41] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755. **2020**.

[42] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341. **2021**.

[43] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11562–11572. **2021**.

[44] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958. **2022**.

[45] Zineng Tang, Jie Lei, and Mohit Bansal. Decembert: Learning from noisy instructional videos via dense captions and entropy minimization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2415–2426. **2021**.

[46] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *International Conference on Learning Representations*. **2021**.

[47] Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang. Hit: Hierarchical transformer with momentum contrast for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11915–11925. **2021**.

[48] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, pages 214–229. Springer, **2020**.

[49] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems*, 33:25–37, **2020**.

[50] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, **2021**.

[51] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738. **2021**.

[52] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once-multi-modal fusion transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20020–20029. **2022**.

[53] Valentin Gabeur, Arsha Nagrani, Chen Sun, Karteek Alahari, and Cordelia Schmid. Masking modalities for cross-modal video retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1766–1775. **2022**.

[54] Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. Teachtext: Crossmodal generalized distillation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11583–11593. **2021**.

[55] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963. **2022**.

[56] Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Object-aware video-language pre-training for retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322. **2022**.

[57] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787. **2014**.

[58] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. **2013**.

[59] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598. **2018**.

[60] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 754–763. **2017**.

[61] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7386–7395. **2018**.

[62] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):765–779, **2018**.

[63] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863. **2022**.

[64] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555. **2019**.

[65] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8085–8095. **2021**.

[66] Yuhan Shen and Ehsan Elhamifar. Semi-weakly-supervised learning of complex actions from instructional task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354. **2022**.

[67] Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. Action duration prediction for segment-level alignment of weakly-labeled videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2053–2062. **2021**.

[68] Zijia Lu and Ehsan Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 19903–19913. **2022**.

[69] Mikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems*, 34:13782–13793, **2021**.

[70] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916. **2022**.

[71] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. What you say is what you show: Visual narration detection in instructional videos. *arXiv preprint arXiv:2301.02307*, **2023**.

[72] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376. **2018**.

[73] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074. **2019**.

[74] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6341–6350. **2019**.

[75] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 557–573. Springer, **2020**.

[76] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165. **2021**.

[77] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761. **2018**.

[78] Wei Chen, Caiming Xiong, Ran Xu, and Jason J Corso. Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 748–755. **2014**.

[79] Limin Wang, Yu Qiao, Xiaoou Tang, and Luc Van Gool. Actionness estimation using hybrid fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2708–2717. **2016**.

[80] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Fine-grained temporal contrastive learning for weakly-supervised temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19999–20009. **2022**.

[81] Linjiang Huang, Liang Wang, and Hongsheng Li. Multi-modality self-distillation for weakly supervised temporal action localization. *IEEE Transactions on Image Processing*, 31:1504–1519, **2022**.

[82] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11320–11327. **2020**.

[83] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 420–437. Springer, **2020**.

[84] Pilhyeon Lee and Hyeran Byun. Learning action completeness from points for weakly-supervised temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13648–13657. **2021**.

[85] Sanath Narayan, Hisham Cholakkal, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13608–13617. **2021**.

[86] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Zheng Shou. Augmented transformer with adaptive graph for temporal action proposal generation. In *Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis*, pages 41–50. **2022**.

[87] Ziyi Liu, Le Wang, Wei Tang, Junsong Yuan, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through learning explicit subspaces for action and context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2242–2250. **2021**.

[88] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5502–5511. **2019**.

[89] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, and Guangwei Yu. Weakly supervised action selection learning in video. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 7587–7596. **2021**.

[90] Jiawei Chen, Yin Cui, Guangnan Ye, Dong Liu, and Shih-Fu Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *Proceedings of international conference on multimedia retrieval*, pages 1–8. **2014**.

[91] Nazli Ikizler-Cinbis, R Gokberk Cinbis, and Stan Sclaroff. Learning actions from the web. In *2009 IEEE 12th International Conference on Computer Vision*, pages 995–1002. IEEE, **2009**.

[92] Nazli Ikizler-Cinbis and Stan Sclaroff. Web-based classifiers for human action recognition. *IEEE transactions on multimedia*, 14(4):1031–1045, **2012**.

[93] Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 1338–1345. IEEE, **2012**.

[94] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Attention transfer from web images for video recognition. In *Proceedings of the 25th ACM international conference on multimedia*, pages 1–9. **2017**.

[95] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. **2015**.

[96] Jianguang Zhang, Yahong Han, Jinhui Tang, Qinghua Hu, and Jianmin Jiang. Semi-supervised image-to-video adaptation for video action recognition. *IEEE transactions on cybernetics*, 47(4):960–973, **2016**.

[97] Feiwu Yu, Xinxiao Wu, Jialu Chen, and Lixin Duan. Exploiting images for video recognition: heterogeneous feature augmentation via symmetric adversarial learning. *IEEE Transactions on Image Processing*, 28(11):5308–5321, **2019**.

[98] Yang Liu, Zhaoyang Lu, Jing Li, Tao Yang, and Chao Yao. Deep image-to-video adaptation and fusion networks for action recognition. *IEEE Transactions on Image Processing*, 29:3168–3182, **2019**.

[99] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 670–688. Springer, **2020**.

[100] Andrew Kae and Yale Song. Image to video domain adaptation using web supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 567–575. **2020**.

[101] Jin Chen, Xinxiao Wu, Yao Hu, and Jiebo Luo. Spatial-temporal causal inference for partial image-to-video adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1027–1035. **2021**.

[102] Wei Lin, Anna Kukleva, Kunyang Sun, Horst Possegger, Hilde Kuehne, and Horst Bischof. Cycda: Unsupervised cycle domain adaptation to learn from image to video. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 698–715. Springer, **2022**.

[103] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321. **2018**.

[104] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, **2011**.

[105] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, **2020**.

[106] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, **2019**.

[107] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18952–18961. **2023**.

[108] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10727–10738. **2023**.