UNIVERSITY OF CALIFORNIA
RIVERSIDE

Near-Future Prediction in Videos: Applications in Video Annotation and Frame Reconstruction

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Tahmida B. Mahmud

September 2019

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Matthew Barth
Dr. Evangelos Papalexakis

The Dissertation of Tahmida B. Mahmud is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

The work presented in this thesis would not have been possible without the guidance, inspiration, and support of a number of wonderful individuals and I express my sincere gratitude to all of them for being a part of this amazing journey. First and foremost, I would like to thank my advisor Dr. Amit K. Roy-Chowdhury for his constant motivation and support during the course of this dissertation. He always encouraged me to come up with simple solutions to complex novel problems which have meaningful impact in the research community. I had the freedom to work on different interesting problems, have free discussions on them, and he constantly guided me to come up with efficient solutions through his valuable feedback. I have learned a lot from his expertise and immense knowledge. I feel extremely fortunate and privileged to work under his supervision.

I would also like to express my heartfelt gratitude to my dissertation committee members, Dr. Matthew Barth, Dr. Ertem Tuncel and Dr. Evangelos Papalexakis for giving me valuable feedback and constructive comments in improving the quality of this dissertation. I have learned a lot from the courses offered by Dr. Tuncel and Dr. Barth. Special thanks are reserved for my undergrad advisor Dr. Md. Kamrul Hasan from Bangladesh University of Engineering and Technology for nurturing me as a researcher as an undergraduate, and instilling in me the curiosity to pursue PhD. I also owe a lot to all my internship mentors Natalia Vassilieva, Sergey Serebryakov from Hewlett Packard Labs, and Joseph Tighe from Amazon.com Services, for their support and encouragement.

I would like to thank my fellow labmates in the Video Computing Group at UC Riverside. Dr. Mahmudul Hasan has simultaneously been a mentor and co-author, and I am grateful for his valuable insights and feedback over the years. My special thanks to Dr. Anirban Chakraborty, Dr. Jawadul Hasan, Dr. Rameswar Panda, Dr. Niluthpol Chowdhury, Sujoy Paul, and Abhishek Aich for

the intellectual discussions we have had in the last five years.

I would like to thank the NSF, and ONR for their grants to Dr. Roy-Chowdhury, which partially supported my research. I thank Victor Hill for setting up the computing infrastructure used in most of the works presented in this thesis and for being prompt with any kind of maintenance issues.

I am grateful to my family, friends, and acquaintances who remembered me in their prayers. My friends at UCR deserve special mention as they have been like my family far away from home. I especially thank my childhood friend Farah Naz Taufiq with whom I shared the ups and downs of my life as a PhD researcher. Last but not the least, I would like to thank my mother Selina Akhtar and father Mahmud Hossain for their inspiration and continuous support to pursue my PhD. My gratefulness for their sacrifice cannot be expressed in words. I am thankful to my late grandfather Chowdhury Hafizur Rahman who has always been my greatest source of inspiration. Most importantly, I wish to thank my loving and supportive husband, Mohammad Billah, for constantly supporting me in every possible way so that I could pay attention to the studies only and achieve my objective without any obstacle on the way. His eternal support, love, and understanding of my goals and aspirations have always been my greatest strength.

Acknowledgment of previously published materials: The text of this dissertation, in part or in full, is a reprint of the material as appeared in four previously published/submitted papers that I first authored. The co-author Dr. Amit K. Roy-Chowdhury, listed in all four publications, directed and supervised the research which forms the basis for this dissertation. The papers are as follows.

1. Tahmida Mahmud, Mahmudul Hasan, and Amit K. Roy-Chowdhury, “Prediction of Activity Labels and Starting Times in Untrimmed Videos”, IEEE Conference on Computer Vision

(ICCV) 2017.

2. Tahmida Mahmud, Mohammad Billah, Mahmudul Hasan, and Amit K. Roy-Chowdhury, “Captioning Near-Future Activity Sequences”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018 (Under Review).

3. Tahmida Mahmud, Mohammad Billah, and Amit K. Roy-Chowdhury, “Multi-View Frame Reconstruction with Conditional GAN”, IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2018.

4. Tahmida Mahmud, Mahmudul Hasan, and Amit K. Roy-Chowdhury, “Exploiting Early Prediction for Scalable Video Annotation”, IEEE Conference on Computer Vision (ICCV) 2019 (Under Review).

To my parents and my husband for all the support.

ABSTRACT OF THE DISSERTATION

Near-Future Prediction in Videos: Applications in Video Annotation and Frame Reconstruction

by

Tahmida B. Mahmud

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, September 2019
Dr. Amit K. Roy-Chowdhury, Chairperson

Near-future prediction in videos has crucial impact on a wide range of practical applications which require anticipatory response. In videos, prediction can be performed in different spaces such as labels, captions and frames. Labels can be predicted for a longer horizon in future but are less informative than frames. Video frames are much richer in content than labels but only a few frames can be predicted ahead. Captions lie in between these two extremes: they can describe changes in activities for a longer prediction horizon and provide a much richer description than labels. In this thesis, we provide three distinct prediction frameworks leveraged upon different computer vision and machine learning techniques. However, these solution methods require lots of labeled data which is challenging due to high annotation cost. Thus, we also propose a novel early prediction framework so that video annotation becomes scalable.

Most of the existing works on labeling human activities focus on the recognition or early recognition problem where complete or partial observations of the activity are available. However, in the prediction problem we are addressing, no observation of the future activity is available beforehand. We propose a system that can infer about the labels and the starting time of a sequence of future unobserved activities combining different context attributes from the observed portion of the video . Next, we propose a sequence-to-sequence learning-based approach using an encoder-decoder LSTM pair for captioning the near-future unobserved activity sequences.

Building upon the prediction framework, we also work on the frame reconstruction problem in a multi-camera scenario. When a camera has multiple missing frames and available frames within the camera are far apart, the corresponding frames from other overlapping cameras become crucial for reconstruction . We propose an adversarial approach using conditional Generative Adversarial Network (cGAN) where the conditional input is the preceding or following frames within the camera or the corresponding frames from other cameras, all of which are merged together using a weighted average. We also propose an adversarial learning solution to the multi-modal frame reconstruction problem where we learn a mapping between 3D LIDAR point clouds and RGB images. This facilitates faster processing since fusion-based approaches which try to combine the advantages from both sources of data consume huge computing resources.

We also consider the video annotation problem, as it crucial for machine learning approaches described above. State-of-the-art video annotation approaches assume that there is no latency for looking up the correct category of label and the annotator is required to watch the whole video segment. However, choosing the correct label from thousands of categories is not instantaneous and the long viewing time adds to the annotation cost. We propose an LSTM-based early prediction framework which can be combined with any existing active learning approach to provide a list of early suggestions to the annotator. This reduces annotation time and cost by a significant margin.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human activity analysis is an active research area in the computer vision community. There are fundamental challenges associated with the problem, such as - the tremendous intra-class variance, large spatio-temporal scale variation, target motion variations, low image resolution, object occlusion, illumination change, viewpoint change etc. Most of the existing works [11, 43, 74, 75, 120, 150] focus on the observed portion of the video. Predicting the future activity labels is critical in real life scenarios, where anticipatory response is required, e.g., video surveillance, human-computer interaction, autonomous navigation, active sensing, video indexing, active gaming, assisted living, etc. However, it is only starting to garner significant interest in the computer vision community. To the best of the our knowledge, our previous work [84] is the only other work in the computer vision community for starting time prediction.

Information from previous activities (sequential activity context) are useful to infer about the activities which follow. Object features or scene context become useful for dealing with ambiguity when there are multiple possible activities. The starting time of the next activity depends on the duration of the last observed activity. To infer about the difference between the starting time of the last observed activity and the future unobserved activity, we use the previous activity features as the inter-activity time context. We develop a deep network which incorporate different context attributes to jointly predict the labels and the starting times of future unobserved activities. The network is trained on the previous activity features and the features of the objects present in the scene.

Next, we focus on providing a richer description of the future activities in the form of captions. Generating description of visual content is an interesting problem in both computer vision and natural language processing community since it exploits the relationship between two of the richest modalities to make semantic representation meaningful. All of the existing works on video captioning [24, 65, 140, 141, 157, 161] focus on the observed portion of the video and ours is the first work which provides captions for a sequence of near-future unobserved activities in videos. Leveraged on our label prediction framework, we start with the labels of the future unobserved activities. Once the labels are available, we map them along with the scene context of the last observed portion to generate captions for future activities using a sequence-to-sequence learning-based approach. We use an encoder-decoder LSTM pair for the mapping task.

The problem of multi-sensor frame reconstruction is closely related with the video prediction problem since it requires information from the previous frame to learn the spatio-temporal representation of the missing frame. Although there have been works on single-view frame reconstruction [15,53,131], to the best of our knowledge, ours is the first work to solve it in a multi-camera scenario. Multi-sensor reconstruction becomes helpful specially when adjacent available frames within the camera are far apart. Motivated by [52], we present an adversarial approach to learn a joint spatio-temporal representation of the missing frame in a multi-camera scenario conditioned on the preceding and following frames within the camera as well as on the corresponding frames in other overlapping cameras using conditional Generative Adversarial Network (cGAN) [89]. All of these representations are then merged together using a weighted average. Multi-modal frame reconstruction is crucial in autonomous navigation applications since fusion-based approaches combining information from multiple sensors are subject to huge consumption of computational resources. We propose a cGAN architecture to learn a mapping between 3D point clouds from mobile terrestrial LIDARS and RGB images from cameras which facilitates scene reconstruction from LIDAR data only.

All of these above mentioned approaches require a large amount of labeled data which adds to high annotation cost. It takes time to look up the correct category of label from thousands of labels and videos can be very long. Video annotation methods need to scale with growing number of video categories and the time spent in watching a video needs to be considered in evaluating the performance of the annotation methods. Motivated by these challenges, we incorporate a novel early prediction framework in an active learning framework to make the annotation task scalable. The most informative queries are initially selected using label propagation on a similarity graph and sent to the annotator for annotation. The same queries are sent to an LSTM-based early prediction network which dynamically provides suggestions to the annotator. The annotator selects the correct labels from the suggestions without watching the entire video. The early prediction model is incrementally updated using these newly labeled instances.

Main Contributions. We address four novel and practical problems in this thesis as follows.

• First, we develop a novel architecture to jointly model the sequential relationships among activities, scene context and inter-activity time context in order to predict the future activity labels as well as their starting times.

• Second, we solve a novel and relevant problem of captioning a sequence of future unobserved activities in a video using a sequence-to-sequence learning-based approach.

• Third, we solve a novel problem of multi-sensor multi-modal frame reconstruction using conditional Generative Adversarial Network (cGAN).

• Fourth, we propose a novel approach for reducing video annotation cost by combining an early prediction network with existing active learning framework. Our method addresses scalability issue for video annotation since it scales quite efficiently with the number of video categories and significantly reduce both the amount of manual labeling and the long watching time of the videos.

Extensive experiments on different benchmark datasets demonstrate that our approaches perform substantially better compared to baselines and state-of-the-art alternative methods.

1.1 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we present our joint prediction framework for activity labels and starting time combining different context information from the observed portion using an LSTM-based deep network. We propose a sequence-to-sequence learning-based approach for captioning near-future activity sequences in Chapter 3 using an encoder-decoder LSTM pair. In Chapter 4, we propose an adversarial approach for multi-sensor frame reconstruction using conditional Generative Adversarial Network (cGAN) where the conditional inputs are the available frames in the camera network. We also propose an adversarial approach for multi-modal frame reconstruction by learning a mapping between 3D point clouds and RGB images. Finally, in Chapter 5, we propose an early prediction framework combined with any active learning framework for scalable video annotation in terms of number of categories and long viewing time. We conclude the thesis in Chapter 6 by providing some future research directions.

Chapter 2

Joint Prediction of Activity Labels and Starting Times in Untrimmed Videos

Abstract

Most of the existing works on human activity analysis focus on recognition or early recognition of the activity labels from complete or partial observations. Predicting the labels of future unobserved activities where no frames of the predicted activities have been observed is a challenging problem, with important applications, which has not been explored much. Associated with the future label prediction problem is the problem of predicting the starting time of the next activity. In this work, we propose a system that is able to infer about the labels and the starting times of future activities. Activities are characterized by the previous activity sequence (which is observed), as well as the objects present in the scene during their occurrence. We propose a network similar to a hybrid Siamese network with three branches to jointly learn both the future

label and the starting time. The first branch takes visual features from the objects present in the scene using a fully connected network, the second branch takes previous activity features using a LSTM network to model long-term sequential relationships and the third branch captures the last observed activity features to model the context of inter-activity time using another fully connected network. These concatenated features are used for both label and time prediction. Experiments on two challenging datasets demonstrate that our framework for joint prediction of activity label and starting time improves the performance of both, and outperforms the state-of-the-arts.

2.1 Introduction

Human activity analysis is a widely studied computer vision problem. The solution to this problem has crucial impact on a wide range of practical applications such as video surveillance, human-computer interaction, autonomous navigation, active sensing, video indexing, active gaming, assisted living, etc. In spite of the enormous amount of research conducted in this area, the problem is still challenging due to the fundamental challenges inherent to the task, such as - the tremendous intra-class variance among the activities, huge spatio-temporal scale variation, target motion variations, etc. Moreover, low image resolution, object occlusion, illumination change and viewpoint change further aggravate these challenges. The majority of the existing works focus on the recognition of observed activities or early recognition of partially observed activities. In other words, they try to answer queries like *what happened before* or *what is happening right now*, whereas predicting the labels of future activities which have not yet been observed is a scarcely explored problem. In [11, 74, 75, 120, 150], by using the word ‘prediction’, these papers basically refer to the early

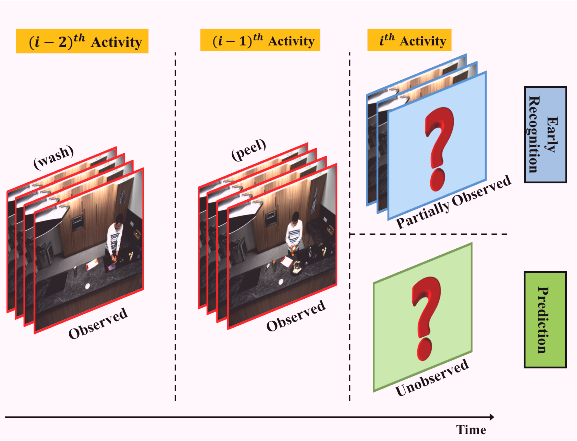


Figure 2.1: An example sequence of a video stream from MPII-Cooking Dataset [117]. Two related problems are explained here - early recognition of the i^{th} activity from partial observations of it, and prediction of its label from previously observed activities only. In the early recognition problem (top-right), the first few frames of the i^{th} activity (cut slices) have been observed. In the prediction problem (bottom-right), no frame of the i^{th} activity has been observed.

recognition task, i.e., predicting the label of the ongoing activity where the first few frames of that activity have already been observed. However, in the prediction problem we are addressing, no observation is available beforehand. The difference between these two problems is illustrated in Figure 2.1. Predicting the future activity labels is critical in real life scenarios, where anticipatory response is required such as active sensing and autonomous navigation. For example, it can help autonomous vehicles to decide how to maneuver depending on the next predicted activity and its time of occurrence, or assist robots to make future decisions. There are only a few approaches [13, 63] which perform label prediction on real-life activity datasets like VIRAT [94]. To the best of our knowledge, only one work [84] in the video analysis community addresses the problem of predicting the starting time of future unobserved activities.

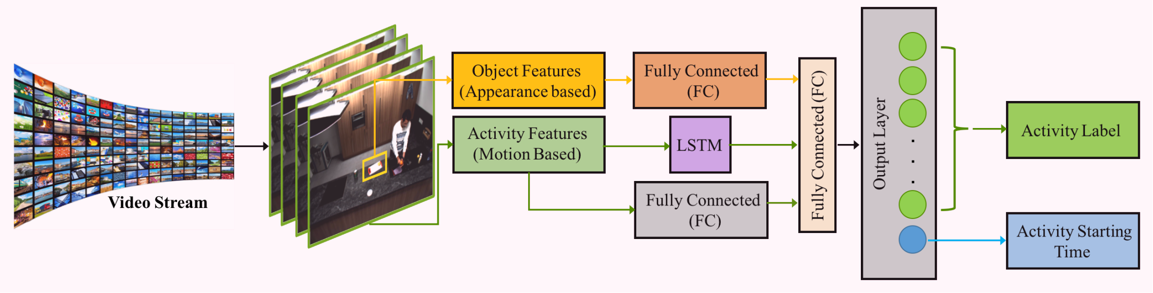


Figure 2.2: Overview of our approach. For joint prediction, both activity features (motion-based) from previous activities and object featues present in the scene are used for training. Please refer to Section 2.3.2 for details.

2.1.1 Overview of the Proposed Approach

In this work, for a video observed up to a particular time, we present an integrated approach that can answer two important questions regarding its unobserved portion: *what will happen next* and *when will it happen*, i.e., we predict the **labels** and the **starting times** of future unobserved activities in both coarse (VIRAT Ground Dataset [94]) and fine grained activity datasets (MPII-Cooking Dataset [117]). We pose this as a joint (label and starting time) prediction task because the problems of predicting the label and the starting time of unobserved activities are closely related and handling them together is intuitive. For example, in MPII-Cooking Dataset, ‘cut slices’ can be followed by two probable activities: ‘spice’ or ‘take out from drawer’. Usually, ‘spice’ takes place immediately after ‘cut slices’; but if there is a delay, then ‘take out from drawer’ happens before.

Detailed overview of our proposed framework is illustrated in Figure 2.2. We developed a deep network by merging three branches: one with two fully connected layers, another with two LSTM layers and the last one with another two fully connected layers. Finally, we add another fully connected layer to the output of this merged network. The two fully connected layers in the first branch are trained on the features of the objects present in the last observed portion of the scene, the LSTM layers are trained on the visual activity features of the previously observed sequential activities to exploit the context of long term sequential dependency and the two fully connected layers in the third branch are trained on the visual activity features of the last observed activity to model the context of inter-activity time based on the last observed activity label. So, the entire network is trained on both the previous activity features and the features of the objects present in the scene. In the output layer, we use the first few (equal to the number of activity classes) nodes as the logistic regression nodes for label prediction and the last node as a regression node for starting time prediction exploiting the concatenated features. The logistic regression nodes assign different probabilities to the future activity labels from which the label with the highest probability is chosen and the regression node provides the inter-activity time between the future activity and the last observed activity from which the starting time of the future activity is obtained. The motivation behind incorporating different context attributes is explained in Section 2.3.1 with ablation study provided in Sections 2.4.3 and 2.4.4. Our **main contribution** is that we propose a novel architecture that jointly models sequential relationships of the activities, scene context and inter-activity time context in order to predict the future activity labels as well as their starting times.

2.2 Related Works

Our work involves the following areas of interest: activity recognition, future activity label prediction, future activity starting time prediction, and Long Short-Term Memory (LSTM) network. We will review some relevant papers from these areas.

Activity Recognition. Activity recognition approaches based on hand-crafted visual features can be divided into three categories: low-level local feature based methods leveraged on interest point [71], mid-level feature based methods leveraged on tracking and pose analysis [88], and high-level semantic attribute based methods [121]. We would like to refer to article [56] and [103] for a comprehensive review of the state-of-the-art approaches. Most of the traditional approaches rely on hand-engineered local features (e.g., STIP, SIFT-3D, HOG-3D, iDT). However, supervised and unsupervised learning of meaningful hierarchical features from deep neural networks (i.e., autoencoder, sparse coding, and convolutional neural networks) have shown huge success over hand-engineered features recently. C3D feature learned with 3D Convolutional Networks is now the state-of-the-art spatio-temporal feature for video and has been shown to achieve best recognition accuracy in activity recognition tasks [135]. Moreover, methods which consider visual context, i.e., the relationships between different activities and objects in the scene, have been successful for recognition. In [159], object and human pose were used as context. In [16] and [69], group context was used for collective activity recognition. In [23, 51, 153], contextual information has been incorporated with deep networks to improve recognition accuracy. Context has also been shown to be useful for learning the models [43].

Future Activity Label Prediction. There have been a few works which predict the label of the future unobserved activity such as approaches using semantic scene labeling [63], Probabilistic Suffix Tree (PST) [74], augmented- Hidden Conditional Random Field (a-HCRF) [155], Markov Random Field (MRF) [13], kernel-based reinforcement learning [49], max-margin learning [68], and deep network [146]. Among these, only [13, 63] perform label prediction, without any observation of the activity to be predicted. In [146], where visual representation of images is predicted and then recognition algorithm is applied, actions can be anticipated only upto one second in the future.

Future Activity Starting Time Prediction. Predicting the starting times of future unobserved activities is a new research problem in the video understanding community. Although, there are some

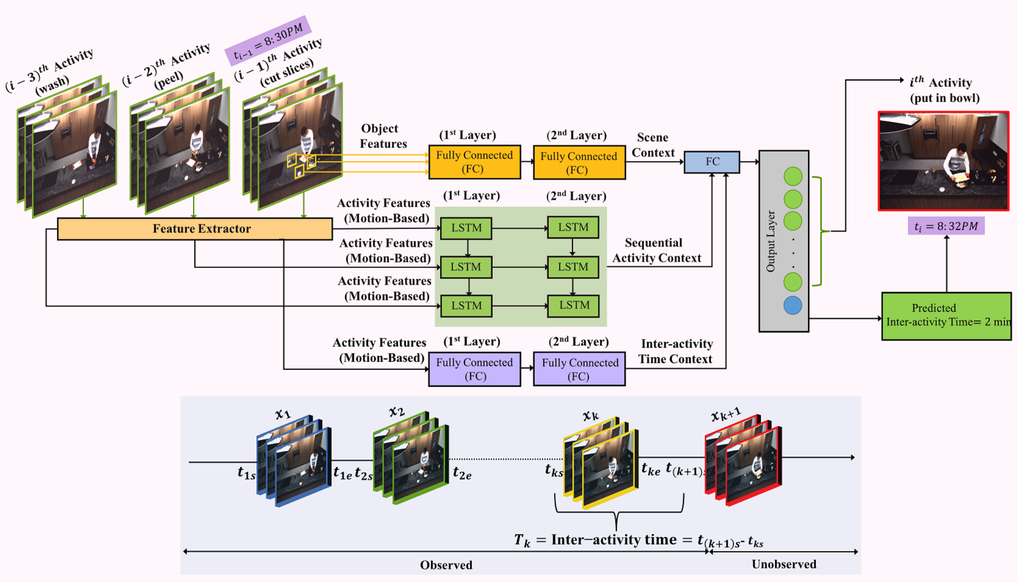

$t_{i-1} = 8{:}30PM$
$(i-3)^{th}$ Activity (wash)
$(i-2)^{th}$ Activity (peel)
$(i-1)^{th}$ Activity (cut slices)
Object Features
(1st Layer)
(2nd Layer)
Fully Connected (FC)
Fully Connected (FC)
Scene Context
FC
i^{th} Activity (put in bowl)
Feature Extractor
Activity Features (Motion-Based)
Activity Features (Motion-Based)
Activity Features (Motion-Based)
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
Sequential Activity Context
Output Layer
$t_i = 8{:}32PM$
Predicted Inter-activity Time= 2 mi
Activity Features (Motion-Based)
(1st Layer)
(2nd Layer)
Fully Connected (FC)
Fully Connected (FC)
Inter-activity Time Context
x_1
x_2
x_k
x_{k+1}
t_{1s}
t_{1e} t_{2s}
t_{2e}
t_{ks}
t_{ke} $t_{(k+1)s}$
T_k = Inter−activity time = $t_{(k+1)s}$- t_{ks}
Observed
Unobserved

Figure 2.3: Proposed architecture for future activity label prediction. The top two fully connected layers (yellow) incorporate the scene context which use object features as input. The two LSTM layers (green) are used to incorporate the sequential activity context which use motion-based features as inputs. The bottom two fully connected layers (purple) are used to incorporate inter-activity time context which use the last observed activity features (motion-based) as input. There is a fully connected layer (blue) where all these layers are merged together. The output layer (gray) performs the final prediction, where the first few nodes (green) are used as the logistic regression nodes for label prediction and the last node (blue) is used as the regression node for starting time prediction. In the problem description figure (bottom), activities have starting times (t_{1s}, t_{2s}, ..., t_{ks}) and ending times (t_{1e}, t_{2e}, ..., t_{ke}). We want to predict the starting time $t_{(k+1)s}$, of the $(k+1)^{th}$ activity by predicting the inter-activity time T_k.

relevant works [82, 164] in other fields, to the best of the our knowledge, there is only one relevant work [84] in the domain of video analysis which is one of our previous works where we modeled the inter-activity times using a Log-Gaussian Cox Process (LGCP). Our new approach outperforms this baseline model.

Long Short-Term Memory (LSTM) Network. Unlike traditional neural networks, Recurrent Neural Network (RNN) has the capability of allowing information to be passed from one step of the network to the next using the loops inherent to their structure. However, in practice, RNNs cannot handle long-term dependencies, primarily because of the vanishing and exploding gradient problem. To overcome the challenge of handling long-term dependency, a special type of RNN called LSTM (Long Short-Term Memory) was introduced in [48]. LSTMs have achieved impressive performance in different sequence learning problems [24, 39, 102, 132, 145]. Its ability to capture long-range dependencies makes it a perfect tool for long-term context incorporation.

2.3 Methodology

2.3.1 Role of Different Context Attributes

In real life scenarios, it is observed that activities follow fixed temporal sequences. Therefore, previous activities can provide useful information about the upcoming ones which can be referred to as **sequential activity context**. Activities are also characterized by the objects present in the scene during the time of their occurrence which can be referred to as **scene context**. For many activities, predicting the future has multiple plausible options. To deal with this specific ambiguity, we take scene context into account along with the sequential information. Thus combining the information obtained from these two different context attributes (temporal sequence and spatial objects), we infer about future unobserved activities. For example, if three sequential activities in a video are ‘wash objects’, ‘peel’ and ‘cut slices’, then there may be two probable choices for the next activity label: ‘spice’ or ‘put in bowl’ (based on two different training instances). But a bowl present in the scene would increase the possibility of the latter choice. Several research works on activity recognition [16, 23, 51, 69, 153, 159, 168] and prediction [13] have shown significant performance improvement by using such context information which are also known as context-aware approaches. Most of the existing works have graphical model based approaches for context incorporation. However, they are not very suitable to handle the context of long-term dependency. As mentioned before, LSTM is a popular choice for sequential context incorporation. LSTM networks are straightforward to fine-tune end-to-end and can handle sequential data of varying lengths. So, we use LSTM to incorporate sequential activity context. However, for including the scene context, there is no need for handling such sequential dependency and fully connected layers can capture this efficiently.

The inter-activity time between different activities depends on their labels. For example, it is obvious from our experience that ‘peel’ or ‘cut slices’ takes more time than ‘wash objects’. Thus, by observing the previous activity features we can infer about the difference between the starting time of the observed activity and the future activity referred to as **inter-activity time context**.

2.3.2 Overall Framework

Our proposed architecture and the basic idea of the problem are shown in Figure 2.3. For our case, the LSTM is used to solve a sequential input, static output problem. We use the activity features extracted from three (chosen empirically) previously observed activities as the LSTM input. Increasing the sequence length does not improve the prediction accuracy significantly (see Parameter Sensitivity in Section 2.4.3 for details). We use a two-layer (chosen empirically) LSTM in the second branch with 256 memory units in each layer. The input of the two (chosen empirically) fully connected layers in the first branch are the visual features extracted from the objects present in the scene with 256 nodes in each layer. The input of the two (chosen empirically) fully connected layers in the third branch are the activity features extracted from the last observed activity with 256 nodes in each layer as well. Finally, the outputs from these three branches are tied together and another fully connected layer is added on top of it. The merging combines the effect of different context attributes. In the output layer, the first few (equal to the number of activity classes) nodes are used as the logistic regression nodes for label prediction and the last node is used as a regression node for starting time prediction.

2.3.3 Model Training Approach

We use the popular open source deep learning package Keras [17] with TensorFlow [1] in the backend which has ready-to-use implementations of LSTM and fully connected layers. The network is trained on a NVIDIA Tesla K40 GPU. The input sequences for the LSTM are chosen in a sliding window manner with a stride of one for data augmentation. For example, to predict the i^{th} activity label, activity features extracted from the $(i-1)^{th}$, $(i-2)^{th}$ and $(i-3)^{th}$ activities are used and for predicting the $(i+1)^{th}$ activity label, activity features extracted from the i^{th}, $(i-1)^{th}$ and $(i-2)^{th}$ activities are used and so on. We use ReLU activation function for all the fully connected layers. In output layer, we use softmax activation function in the logistic regression nodes for label prediction and ReLU activation function in the regression node for starting time prediction. The parameters of the entire network (the LSTM and the fully connected layers) are jointly optimized.

We take the summation of the following two losses to compute the final loss. One is the cross-entropy loss function which is defined as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{X},\mathbf{Y}) &= -\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbf{1}(y^{(i)}=j) \\ &\quad \times \log p(y^{(i)}=j|\mathbf{x}^{(i)}) \end{aligned} \tag{2.1}$$

Here, $\mathbf{X}=\{\mathbf{x}^{(1)},...,\mathbf{x}^{(n)}\}$ is the set of input feature vectors in the training dataset, $\mathbf{Y}=\{y^{(1)},...,y^{(n)}\}$ is the corresponding set of labels for those input features, and $j=\{1,...,c\}$ is the set of class labels. $\mathbf{1}(.)$ is an identity function. For a particular training instance, $\mathbf{x}^{(i)}$ represents the sequential activity features extracted from the previous three activities and the object features from the last observed portion of the scene.

Another is the mean squared loss function which is defined as follows:

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \frac{1}{n} \sum_{i=1}^{n} (q^{(i)} - \hat{q}^{(i)})^2 \tag{2.2}$$

Here, $\mathbf{P} = \{\mathbf{p}^{(1)}, ..., \mathbf{p}^{(n)}\}$ is the set of input feature vectors in the training dataset, and $\mathbf{Q} = \{q^{(1)}, ..., q^{(n)}\}$ is the corresponding set of inter-activity times. $\hat{q}^{(i)}$ represents the predicted inter-activity time given input $p^{(i)}$ where the ground truth inter-activity time is $q^{(i)}$. For a particular training instance, $\mathbf{p}^{(i)}$ represents the activity features extracted from the last observed activity.

To optimize the network, we use a stochastic gradient descent with an adaptive sub-gradient method (Adam) [61] which is popular for its strong theoretical convergence guarantee and impressive history of empirical success. We also tested with Adagrad [28], Adamax [61], Nadam [25] and RMSProp [133] but empirically chose Adam. We use Dropout layer [128] with a probability of 0.2 after each layer to prevent overfitting. We use a batch size of 128 and a learning rate of 0.001. Our network converges roughly at 60 epochs.

2.4 Experiments

We conduct experiments on two challenging datasets: MPII-Cooking Dataset [117] (fine grained indoor activities) and VIRAT Ground Dataset [94] (coarse outdoor activities) to evaluate the performance of our proposed framework.

2.4.1 Datasets

MPII-Cooking Dataset. MPII-Cooking Dataset is a fine grained complex activity dataset where the participants interact with different tools, ingredients and containers to complete a recipe. It has 65 different cooking activities recorded from 12 participants. In total there are 44 videos with a length of more than 8 hours. The dataset contains a total of $5,609$ annotations [117].

VIRAT Ground Dataset. VIRAT Ground Dataset is a challenging human activity dataset which consists of 11 different activities recorded in natural outdoor scenes with background clutter. There are total 329 videos with a length of around 5 hours [94]. However, we use only 275 of them as some videos have incomplete annotations.

Detailed description of these datasets is available in the supplementary material. These datasets are untrimmed and have context information unlike the trimmed datasets popularly used for recognition tasks in activity analysis.

2.4.2 Features

For MPII-Cooking Dataset, we use the bag-of-word based Motion Boundary Histograms (MBH) [20] as activity features. According to [149], these features are extracted around densely sampled points and a codebook is generated using k-means clustering for these 4000 words long features. Scene context features (dimension of 212: 41 for tools, 117 for ingredients and 54 for containers) naturally exist in the dataset. For VIRAT Ground Dataset, we use C3D features [135] as activity features. Scene context features naturally exist in VIRAT Ground Dataset too. We use MBH features for MPII-Cooking Dataset as these features come with the dataset. For VIRAT Ground Dataset, we extract the C3D features as it does not come with any features. We report results for

MPII-Cooking Dataset using C3D features as well.

2.4.3 Label Prediction Results

Objective. The main objective of these experiments is to analyze how well our framework can predict the labels of future unobserved activities.

Performance Measures. The evaluation metrics we use are: 1. multi-class precision (Pr), 2. multi-class recall (Rc), and 3. overall accuracy for top-1 match, top-2 matches and top-3 matches. For all these metrics, the higher value indicates better prediction performance.

Compared Methods. We compare our approach to different state-of-the-art methods. There is no existing method for predicting future activity labels for MPII-Cooking Dataset. Therefore, we compare with a recent recognition approach which estimates the labels of the *observed* activities using a combination of CNN and LSTM [92]. For VIRAT Ground Dataset, there is an existing graphical model based approach [13] and a semantic scene labeling based approach [63]. We compare our method with [13] but cannot compare with [63] since they use scene specific customized set of labels which are not annotated in the dataset. We also compare with a state-of-the-art active learning based recognition approach which uses sparse autoencoder [42] and achieve higher accuracy.

Experimental Setup. For MPII-Cooking Dataset, we use five fold leave-one-person-out cross validation approach for the training-testing split and average our results over these five combinations. Among 12 subjects, we use 7 for training and 5 for testing. For each of the five training instances, we use 7 training subjects and 4 testing subjects for training, leaving 1 from that set for testing. This is done 5 times leaving 1 testing subject out and then the results are averaged. For VIRAT Ground Dataset, we use the first 170 videos for training and the rest for testing.

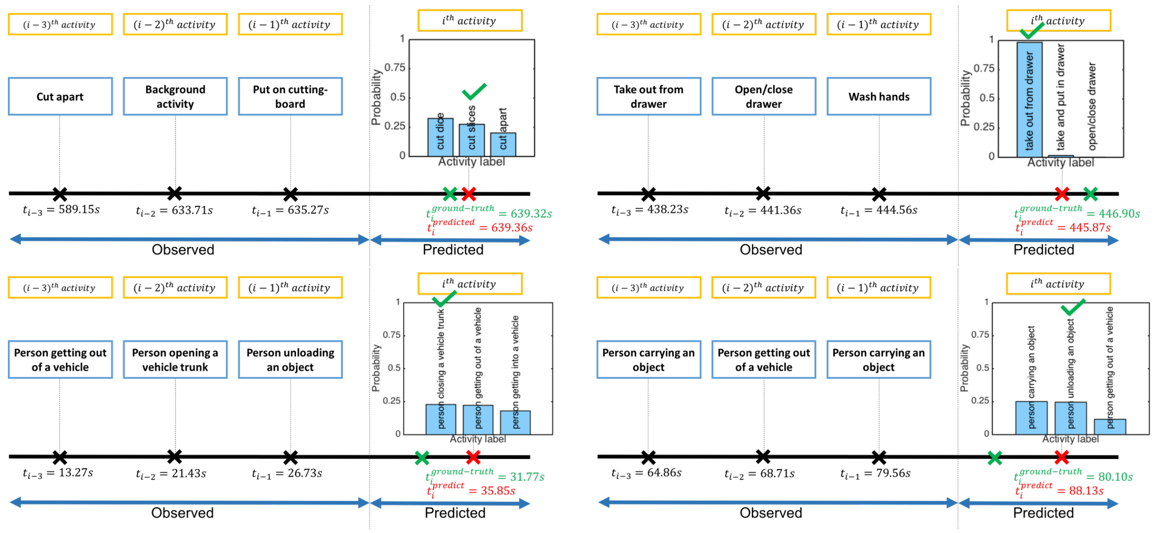


Figure 2.4: Four example activity sequences showing our label prediction results and time prediction results on MPII-Cooking Dataset (top row) and VIRAT Ground Dataset (bottom row). For time prediction, green × marks the ground truth starting time of the activity we are trying to predict, and red × marks the predicted time. For label prediction, top-3 matches are shown here and in most of the cases our top-1 match corresponds to the activity that actually happened (green tick).

Results for MPII-Cooking Dataset. Comparison of our label prediction results on MPII-Cooking Dataset with state-of-the-art method is shown in Table 2.1. The method we compare to did not report all of the evaluation metrics we use- hence the missing values. It is seen that our method outperforms the recognition method proposed in [92]. This is not surprising because in recognition problems the network has to decide among all the activity classes whereas in the sequence learning based prediction task, the network needs to consider only a subset of classes which occurred in the training phase after that particular sequence. Using C3D features, we achieve Top-1 accuracy of 79.9%. The coherence in Top-1 accuracies using both MBH and C3D features indicates that our method is independent of any particular choice of feature.

Results for VIRAT Ground Dataset. Comparison of our label prediction results on VIRAT Ground Dataset with state-of-the-art methods is shown in Table 2.1. It is seen that our method outperforms

MPII-Cooking Dataset [117]	Goal	Precision	Recall	Accuracy % (Top-1)	Accuracy % (Top-2)	Accuracy % (Top-3)
CNN + LSTM [92]	Recognition	34.8	51.7	-	-	-
Proposed Method	Prediction	70.7	66.5	80.1	90.0	93.7
VIRAT Ground Dataset [94]	Goal	Precision	Recall	Accuracy % (Top-1)	Accuracy % (Top-2)	Accuracy % (Top-3)
Sparse Autoencoder [42]	Recognition	-	-	54.2	-	-
Graphical Model [13]	Prediction	-	-	68.5	-	-
Proposed Method	Prediction	49.6	22.2	71.8	79.8	86.4

Table 2.1: Label prediction performance comparisons for MPII-Cooking Dataset and VIRAT Ground Dataset.

the prediction method proposed in [13]. We also achieve higher accuracy than the recognition method proposed by [42]. The intuition behind prediction accuracy being higher than recognition accuracy is explained above. However, for datasets like VIRAT Ground Dataset, where the number of classes is small, prediction accuracy is closer to recognition accuracy. Figure 2.4 depicts some example sequences showing both of our label prediction results and time prediction results on the two datasets.

Multiple Possibilities for Future Activity Label. One particular activity sequence can have multiple possible outcomes. For example, ‘wash objects’ and ‘peel’ can be followed by either ‘cut apart’ and ‘cut slices’. As the network has been trained on both of these possible sequences (in one case the network has probably seen ‘cut apart’ as the next activity and in another case ‘cut slices’ as the next activity), it is hard to say precisely which is the next activity. Earlier we mentioned that in case of multiple possibilities, such as while choosing between ‘spice’ or ‘put in bowl’ after ‘wash objects’, ‘peel’ and ‘cut slices’, a bowl in the scene increases the probability of the activity label being the latter one. But in these types of closely related activities (‘cut apart’ and ‘cut slices’), scene context cannot contribute much as both of the activities require a knife. This is why we present the top-3 choices with the associated probabilities for each of them. We did not go beyond top-3 because after that the probabilities become much lower as we found empirically. This is shown in the first example of Figure 2.4 where our network assigns almost equal probability to all of the possible future activities (‘cut dice’, ‘cut slices’, ‘cut apart’) but the activity which actually happened (‘cut slices’) is the one with the second highest probability. In spite of having these closely related ambiguous activities in the dataset, our top-1 match outperforms the baseline in terms of accuracy. Our method can also handle the case of predicting an unknown label (never seen in training) when the probability of none of the predicted future activities crosses a threshold.

Parameter Sensitivity. We empirically choose a sequence length of 3 for preceding activity features as sequence length of 2, 5, 7 and 9 give relatively lower accuracy for MPII-Cooking Dataset as shown in Table 2.2.

Top-1 Accuracy %				
Sequence Length 2	Sequence Length 3	Sequence Length 5	Sequence Length 7	Sequence Length 9
78.8	80.1	79.2	77.8	77.2

Table 2.2: Parameter sensitivity analysis for MPII-Cooking Dataset.

Ablation Study. Using only sequential activity context and scene context (eliminating inter-activity time context), we get relatively lower label prediction accuracy for MPII-Cooking Dataset than that of our proposed network. Similarly, using only sequential activity context and inter-activity time context (eliminating scene context), we get lower label prediction accuracy than that of our proposed network for MPII-Cooking Dataset. These ablation study results shown in Table 2.3 justifies the integration of label and time prediction.

Top-1 Accuracy %			
Dataset	Proposed Network	Removing Inter-activity Time Context	Removing Scene Context
MPII-Cooking [117]	80.1	75.1	33.1
VIRAT Ground [94]	71.8	69.2	61.0

Table 2.3: Ablation study for label prediction for both of the datasets.

2.4.4 Starting Time Prediction Results

Objective. The main objective of these experiments is to analyze how well our framework can predict the starting times of future unobserved activities.

Performance Measures. We use Root-Mean-Square Error (RMSE) as our evaluation metric. The lower the value, the better is the prediction performance.

Compared Method. We compare our approach to state-of-the-art starting time prediction method (a statistical model) [84]. In [84], there is an underlying assumption of exponential distribution for the inter-activity time. Our new approach is free from this assumption.

Experimental Setup. For experiments on MPII-Cooking Dataset, we use five fold leave-one-person-out cross validation approach for the training-testing split and average our results over these five combinations. For experiments on VIRAT Ground Dataset, we use the first 210 videos for training and the rest of them for testing.

Results for MPII-Cooking Dataset. Comparison of our starting time prediction results on MPII-Cooking Dataset with state-of-the-art method is shown in Table 2.4. It is seen that our method outperforms [84]. We also analyze our time prediction result as a function of the last observed activity label and as a function of the label of the activity being predicted. Figure 2.5 shows the RMSE values based on the label of the last observed activity (top) and the label of the predicted activity (bottom) for MPII-Cooking Dataset. It is seen that only one of the observed activity labels

MPII-Cooking Dataset [117]	Goal	Average Inter-activity Time (sec)	Average RMSE (sec)
Statistical Model [84]	Prediction	5.3426	3.9431
Proposed Method	Prediction	5.3426	1.2454
VIRAT Ground Dataset [94]	Goal	Average Inter-activity Time (sec)	Average RMSE (sec)
Proposed Method	Prediction	13.9567	10.4560

Table 2.4: Starting prediction performance comparisons for MPII-Cooking Dataset and VIRAT Ground Dataset.

(28) (top) and some of the predicted activity labels (bottom) are contributing to a higher amount of error. We found that if the last observed activity is a relatively longer one by nature, such as 'make puree' (label 28 in Figure 2.5 (top)), then the predicted starting time of the next unobserved activity is relatively more erroneous.

Results for VIRAT Ground Dataset. Our starting time prediction result on VIRAT Ground dataset is shown in Table 2.4. The state-of-the-art starting time prediction method [84] does not have results on this dataset. For VIRAT Ground Dataset, there are randomly occurring artificial gaps between many activities. There is no way to train a system to predict the starting time of the next activity with such gaps, since there is no underlying structure in them. (Note that label prediction still works because there is structure in what an actor does next, just not when). Thus, we identify activity sequences where there is a regular pattern of activities happening one after another and show results only on them. For example, labels like 'person loading an object', 'person unloading an object', 'person opening a vehicle trunk', 'person closing a vehicle trunk' belong to natural sequences where we can predict when the next activity will happen. As explained above, while suitable for the label prediction problem given the continuous nature of the data, this dataset is not ideal for activity starting time prediction analysis, which, we believe, is making the error higher here.

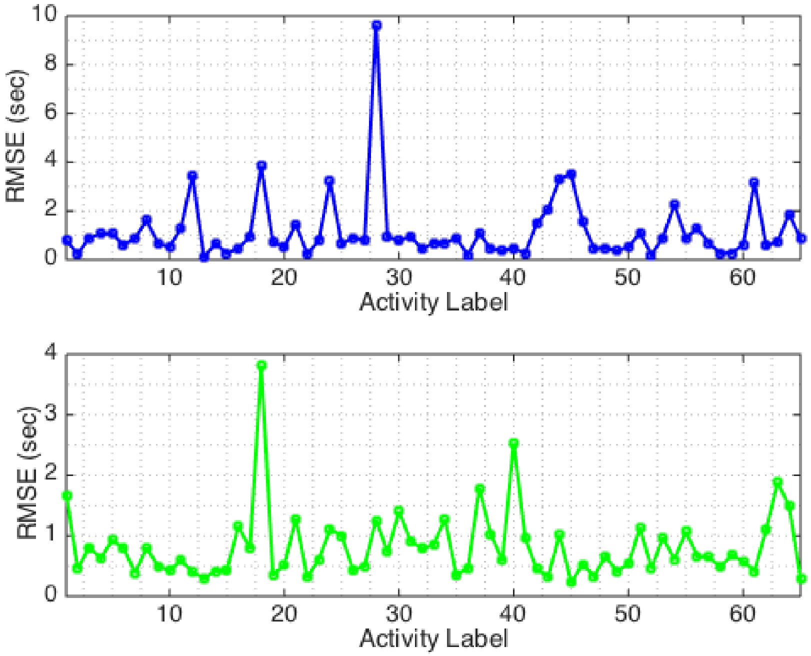


Figure 2.5: RMSE values based on the label of the observed activity (top) and the label of the predicted activity (bottom) for MPII-Cooking Dataset.

Ablation Study. Using only inter-activity time context (eliminating sequential activity context and scene context), we get a higher RMSE for starting time prediction than that of our proposed network for MPII-Cooking Dataset. This ablation study result shown in Table 2.5 justifies the integration of label and time prediction.

Average RMSE (sec)	
Proposed Network	Removing Activity Context & Scene Context
1.2454	1.4872

Table 2.5: Ablation study for starting time prediction for MPII-Cooking Dataset.

2.4.5 Effect on Prediction Horizon

For label prediction, we perform multi-step prediction where we predict the next-to-next activity i.e., 2-step prediction (using activity features from the $(i-3)^{th}$, $(i-2)^{th}$ and $(i-1)^{th}$ activities, we predict the label of the $(i+1)^{th}$ activity) and the next-to-next-to-next activity (3-step prediction). As expected, the accuracy decreases as the prediction horizon increases. For starting time prediction, we also perform multi-step prediction. For example, for 2-step prediction, we train our model using the features of the $(i-1)^{th}$ activity, and its inter-activity time with the $(i+1)^{th}$ activity. During testing, we use the observed features to predict the starting times of the next-to-next activities. As the prediction horizon increases, there is a gradual accumulation of error. The decrease in accuracy for multi-step label prediction for both of the datasets and the increase in RMSE for multi-step starting time prediction for MPII-Cooking Dataset are shown in Figure 2.6.

We did not perform multi-step starting time prediction on VIRAT Ground Dataset because of the random gaps between activities as explained earlier. We did not go beyond 3-step for joint prediction as the RMSE error for starting time prediction is already quite high for 3-step prediction shown in Figure 2.6. However, when we do label prediction separately as an ablation study for prediction horizon, i.e., using a network with only sequential activity context and scene context, the label prediction results upto 5-step prediction for both of the datasets are shown in Figure 2.7 averaged across all of the activity labels. These demonstrate that joint estimation of activity label and starting time leads to higher accuracy, but comes at the cost of a shorter forecasting horizon.

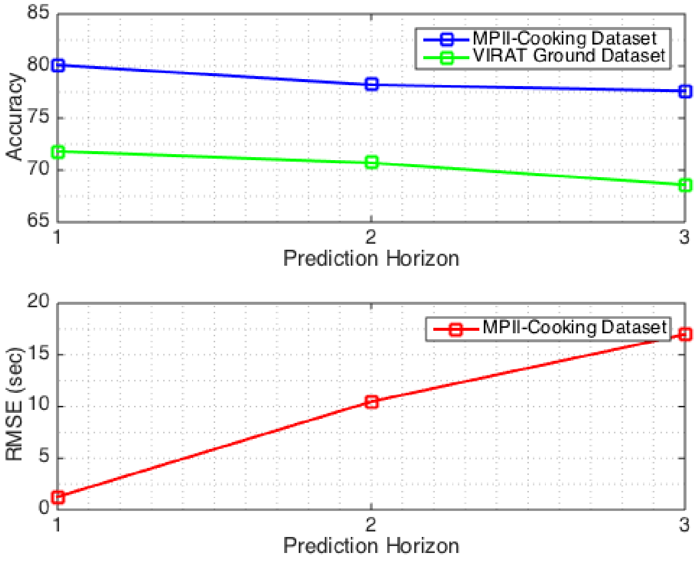


Figure 2.6: Accuracy of the predicted labels (top) and RMSE of the predicted starting times (bottom) for multi-step prediction. For both of the datasets, the label prediction accuracy decreases and for MPII-Cooking Dataset, the RMSE for predicted times increases with the increasing forecasting horizon as expected.

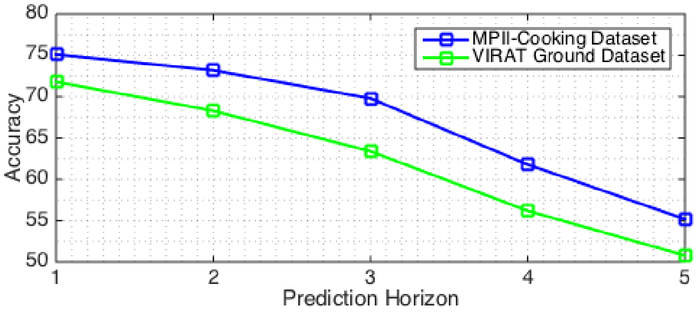


Figure 2.7: Accuracy of the predicted labels for multi-step prediction without inter-activity time context. For both of the datasets, the label prediction accuracy decreases as we try to predict further ahead as expected.

2.5 Conclusions

In this work, we propose a framework for jointly predicting the label and the starting time of future unobserved activity by taking advantage of the combination of LSTM and fully connected layers to exploit the contextual relationship among activities and objects. Rigorous experimental analysis on two challenging datasets proves the robustness of our framework. Our approach is capable of both multi-step label prediction and multi-step time prediction with reasonable error. In future, we plan to extend our prediction method for multi-camera environment and investigate how to predict new unseen activity classes.

Chapter 3

Captioning Near-Future Activity Sequences

Abstract

Most of the existing works on human activity analysis focus on recognition or early recognition of the activity labels from complete or partial observations. Similarly, existing video captioning approaches focus on the observed events in videos. Predicting the labels and the captions of future activities where no frames of the predicted activities have been observed is a challenging problem, with important applications that require anticipatory response. In this work, we propose a system that can infer about the labels and the captions of a sequence of future activities. Our proposed network for label prediction of a future activity sequence is similar to a hybrid Siamese network with three branches where the first branch takes visual features from the objects present in the scene, the second branch takes observed activity features and the third branch captures the last

observed activity features. The predicted labels and the observed scene context are then mapped to meaningful captions using a sequence-to-sequence learning based method. Experiments on three challenging activity analysis datasets and a video description dataset demonstrate that our label prediction framework for a future activity sequence outperforms the state-of-the-art and we achieve comparable performance with the state-of-the-art video captioning approaches for observed events.

3.1 Introduction

Activity analysis is a widely studied problem in the computer vision community. Most of the existing works focus on recognition of observed activities or early recognition of partially observed activities. Predicting the labels of future activities which have not yet been observed is a scarcely explored problem and different from the recognition problem, where inferences need to be made on activity features which have been observed. The word ‘prediction’ has been used in [11, 74, 75, 120, 150], referring to the early recognition task, i.e., predicting the label of the ongoing activity where the first few frames have already been observed. However, in the prediction problem we are addressing, *no observation is available beforehand*. Predicting the future activity labels is critical in real life scenarios, where anticipatory response is required based on an observed segment of the video, e.g., driver intent prediction [90, 169] in Advanced Driver Assistance Systems (ADAS) where a description of which lane the driver might move into in the near future is necessary to predict the likelihood of potential collisions in complex traffic scenarios, or Human Intent Prediction (HIP) [86, 134] in human-robot collaboration where the robot may need to predict what the human may do in the future to ensure safety and efficiency. There are only a few approaches [13, 63] which perform label prediction on real-life activity datasets likes VIRAT [94]. There is only one work

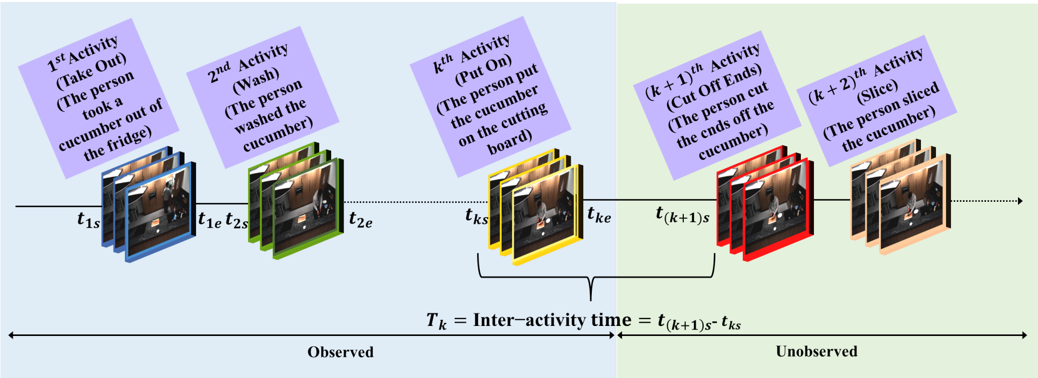

1^{st} Activity (Take Out) (The person took a cucumber out of the fridge)
2^{nd} Activity (Wash) (The person washed the cucumber)
k^{th} Activity (Put On) (The person put the cucumber on the cutting board)
$(k+1)^{th}$ Activity (Cut Off Ends) (The person cut the ends off the cucumber)
$(k+2)^{th}$ Activity (Slice) (The person sliced the cucumber)
t_{1s}
t_{1e}
t_{2s}
t_{2e}
t_{ks}
t_{ke}
$t_{(k+1)s}$
T_k = Inter−activity time = $t_{(k+1)s}$- t_{ks}
Observed
Unobserved

Figure 3.1: There are k activities in the observed portion of a video with starting times (t_{1s}, t_{2s}, ..., t_{ks}) and ending times (t_{1e}, t_{2e}, ..., t_{ke}). We want to predict the labels and the captions of $(k+1)^{th}$, $(k+2)^{th}$,... activities.

which perform label prediction for a future sequence of activities [3].

Generating description of visual content is an active research area in both computer vision and natural language processing community. Since vision and language are two of the richest interaction modalities available to humans, it is crucial to understand the relationship between them. Language is the most natural way to make information from any semantic representation meaningful. In the last few years, this problem has received significant attention for image captioning [31, 54, 67, 144] as well as video captioning [5, 21, 24, 40, 58, 65, 66, 118, 140, 141, 161, 163]. Unlike image description, video description has to deal not only with the appearance of the objects but also with motion over time. To the best of our knowledge, all of the existing works on video captioning focus on the observed portion of the video, i.e., describe events which have already happened or happening at the moment. Ours is the first work where we look into the problem of providing captions for a sequence of *near-future unobserved events in videos*. Generating the labels of future unobserved activities can be considered as the first step towards describing the future. But it may be desirable to offer a richer description than a simple one-word/phrase label for specific applications like assistive

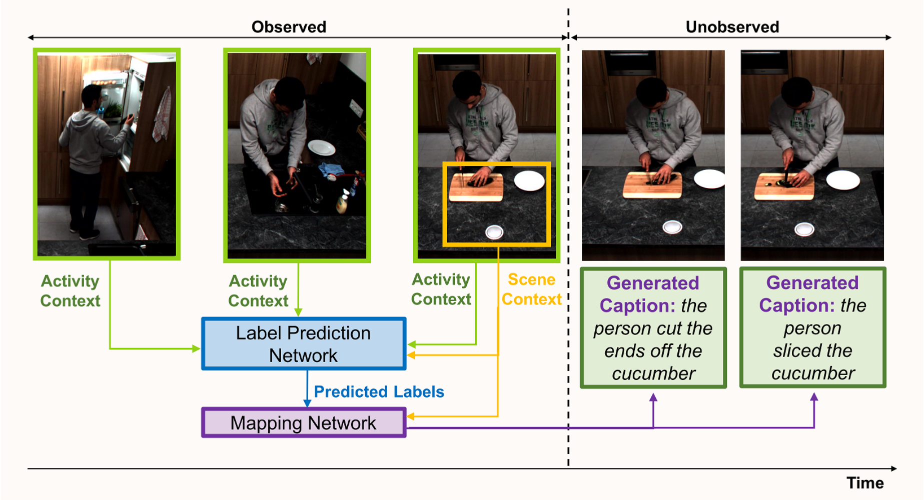


Figure 3.2: Overview of our approach. The label prediction network is trained on both the sequential activity features from previously observed activities and the object features present in the last observed portion of the scene. The sequence-to-sequence learning based mapping network finally maps the sequential labels and observed scene context to a sequence of captions. A detailed version of this figure is given in Fig. 3.3.

systems [30, 97] for the visually impaired. There has been work on generating future frames [146], which are much richer in content, but are constrained to only a few such frames. Our work lies in between these two extremes: it can generate semantically meaningful captions that describe changes in activities and thus able to predict much further in time than the frame generation work [146], while at the same time, provides a much richer description than label prediction [3, 13, 63, 85].

3.1.1 Problem Definition

For a video observed up to a certain time, we want to predict the labels and the captions of the future activity sequence. This is illustrated in Fig. 3.1. We have observed up to the k^{th} activity and want to predict the labels and the captions of the future activity sequence, i.e., the labels and the captions of $(k+1)^{th}$, $(k+2)^{th}$, $\cdots$ activities and the starting time of that sequence, i.e., $t_{(k+1)s}$.

3.1.2 Overview of the Approach

In this work, we present an integrated approach to answer two important questions regarding the unobserved portion of a video observed up to a particular time: *what activities will happen next*, and *what captions describe them best*. We predict the **labels** of a sequence of future unobserved activities in both coarse (VIRAT Ground Dataset [94]) and fine grained activity datasets (MPII-Cooking Dataset [117] and MPII-Cooking 2 Dataset [119]). This is posed as a joint label and starting time prediction task because intuitively the problem of predicting the label and the starting time of unobserved activities are closely related. For example, in MPII-Cooking Dataset, 'cut slices' can be followed by two probable activities: 'spice' or 'take out from drawer'. Usually, 'spice' takes place immediately after 'cut slices'; but if there is a delay, then 'take out from drawer' happens before. Once the labels are available, we map them along with the scene context of the last observed portion to generate meaningful **captions** for a sequence of future activities. Instead of using a rule- or template-based natural language generation (NLG) approach, we are motivated by the data driven domain-independent learning based approach [132] which replaced rule based methods in statistical machine translation. Instead of performing the mapping between two language spaces, we are doing a mapping from labels to captions. This sequence-to-sequence learning based approach makes minimal assumptions on the sequence structure.

Detailed overview of our proposed framework is illustrated in Fig. 3.2. We develop a deep network by merging three branches: one with two fully connected layers, another with two LSTM layers and the last one with another two fully connected layers. There is another fully connected layer to the output of this merged network. The two fully connected layers in the first branch are trained on the features of the objects present in the last observed portion of the scene, the LSTM layers are trained on the visual activity features of the previously observed three sequential activities to exploit the context of long term sequential dependency and the two fully connected layers in the third branch are trained on the visual activity features of the last observed activity to model the context of inter-activity time based on the last observed activity label. The network is trained on the previous activity features and the features of the objects present in the scene.

In the output layer, for each activity of the future sequence, we use the first few (equal to the number of activity classes) nodes as the logistic regression nodes for label prediction. The logistic regression nodes assign different probabilities to the future activity labels from which the label with the highest probability is chosen. For generating captions for a sequence of future unobserved activities, we use a multi-layered LSTM to map the predicted labels and observed scene context to a fixed dimensional vector. Another deep LSTM (which is conditioned on the input sequence) is used for extracting the target sequence (caption) from that vector. The ability of LSTM layers to incorporate long term sequential dependencies makes it a suitable choice for this application.

3.1.3 Main Contributions

In this work, we propose a deep architectural framework which exploits the context of sequential dependency, the context of the objects present in the scene and the nature of the activities for future activity label prediction and caption generation. The **main contributions** of this work are:

1. We propose a novel architecture that jointly models the sequential relationships of the activities, scene context and the last observed activity features in order to predict the labels of a future activity sequence.

2. We solve a novel and relevant problem of captioning a sequence of future unobserved events of a video using a sequence-to-sequence based learning approach.

3. We perform extensive experiments that show the effectiveness of the proposed framework.

3.2 Related Works

Our work involves the following areas of interest: video captioning, future activity label and caption prediction, and Long Short-Term Memory (LSTM) network. We will review some relevant papers from these areas.

Video Captioning. The initial works on video captioning [5,41,59,60,64,72] focus on rule-based systems where sentences are generated using predefined templates following certain linguistic rules. Later, learning based data driven approaches [21,40,66,116,118,130,158,162] became popular. As the methods started becoming free from manual engineering, the problem became more scalable providing flexibility to work with larger datasets. Recently, Recurrent Neural Network (RNN) based approaches [24,140,141,157,161] have achieved promising performance in video captioning. One of the earliest works [141] using RNNs extends the image captioning methods by average pooling the video frames which only works for short video clips containing just one event. To overcome this shortcoming, recurrent encoder [24], [157], [140] based model and attention model [161] have been proposed. [163] uses a hierarchical RNN to generate a paragraph for richer description. Another paper [65] performs dense-captioning of events in videos using context information. All of them focus on the observed portion of the video only; to the best of our knowledge, there is no existing work which can generate captions for the future unobserved portion of a video.

Long Short-Term Memory (LSTM) Network. Unlike traditional neural networks, Recurrent Neural Network (RNN) has the capability of allowing information to be passed from one step of the network to the next using the loops inherent to their structure. However, in practice, RNNs cannot handle long-term dependencies, primarily because of the vanishing and exploding gradient problem.To overcome the challenge of handling long-term dependency, a special type of RNN called LSTM (Long Short-Term Memory) was introduced in [48]. LSTMs have achieved impressive performance in different sequence learning problems [24, 39, 102, 132, 145]. Its ability to capture long-range dependencies makes it a perfect tool for long-term context incorporation.

Future Activity Label and Caption Prediction. There have been a few works which predict the future unobserved activity such as approaches using semantic scene labeling [63], Probabilistic Suffix Tree (PST) [74], augmented- Hidden Conditional Random Field (a-HCRF) [155], Markov Random Field (MRF) [13], kernel-based reinforcement learning [49], max-margin learning [68], and deep network [3, 85, 114, 146]. Among these, only [3, 13, 63, 85] perform prediction, without any observation of the activity to be predicted, in the label space. In [146], where visual representation of images is predicted and then recognition algorithm is applied, actions can be anticipated only upto one second in the future. The focus of [114] is forecasting behavior/goal where the fundamental state variables involved are different than the label space. There is a recent work [3] which infers about the labels of a future activity sequence using a CNN-based and a RNN-based approach. However, they predict the labels of a future unobserved activity sequence only; whereas the main focus of this work is predicting the captions of a future activity sequence. *Our previous work on activity prediction [85] has achieved the highest accuracy on two challenging activity datasets incorporating different context attributes but did not perform sequence prediction.*

Extension to Previous Works. The goal of this work is to predict the captions of a sequence of future activities which is, to the best of our knowledge, the first work in this area. This is leveraged on our previously published paper on activity label and starting time prediction [85]. Instead of predicting the label of one future activity at a time, here we are predicting the labels of a sequence of future activities and finally captioning the future activity sequence using the predicted label information. We conduct experiments on a new dataset called MPII-Cooking 2 Dataset [119] demonstrating the effectiveness of our captioning method.

3.3 Methodology

In this section, we discuss the motivation behind the choice of our network explaining the importance of different context attributes for the task, the network architecture in details, the training scheme and the way we obtained the final results in the test phase.

3.3.1 Label Prediction for Activity Sequences

Role of Different Context Attributes

Activities follow fixed temporal sequences in real life scenarios. Therefore, previous activities can provide useful information about the upcoming ones which can be referred to as **sequential activity context**. Activities are also characterized by the objects present in the scene during the time of their occurrence which can be referred to as **scene context**. For many activities, predicting the future has multiple plausible options. To reduce this specific ambiguity, we take scene context into account along with the sequential information. Thus combining the information obtained from these two different context attributes (temporal sequence and spatial objects), we infer the

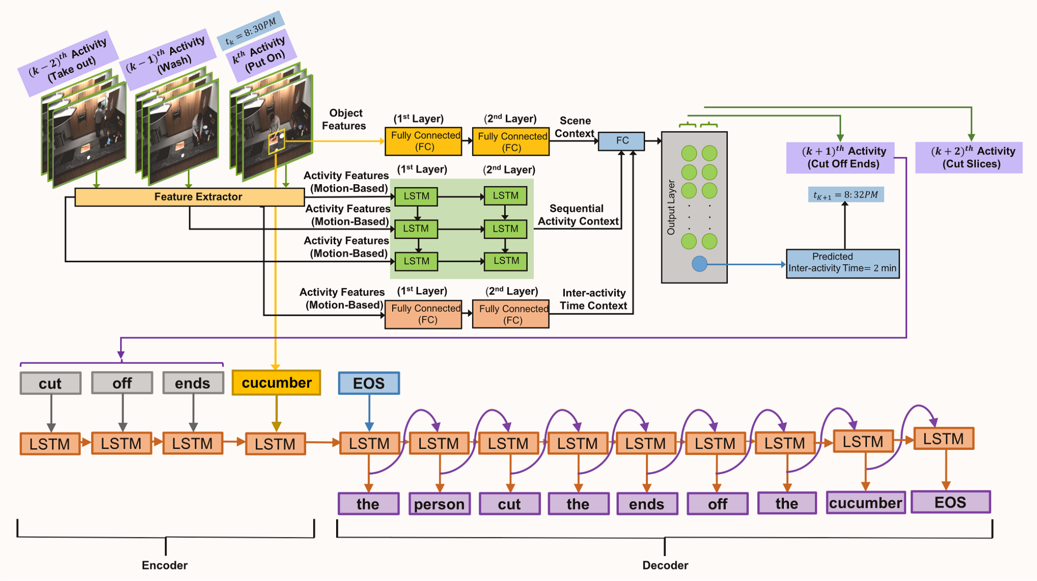

$(k-2)^{th}$ Activity (Take out)
$(k-1)^{th}$ Activity (Wash)
$t_k = 8{:}30PM$
k^{th} Activity (Put On)
Object Features
(1st Layer)
(2nd Layer)
Fully Connected (FC)
Fully Connected (FC)
Scene Context
FC
Feature Extractor
Activity Features (Motion-Based)
Activity Features (Motion-Based)
Activity Features (Motion-Based)
LSTM
Sequential Activity Context
Activity Features (Motion-Based)
Fully Connected (FC)
Fully Connected (FC)
Inter-activity Time Context
Output Layer
$(k+1)^{th}$ Activity (Cut Off Ends)
$(k+2)^{th}$ Activity (Cut Slices)
$t_{K+1} = 8{:}32PM$
Predicted Inter-activity Time= 2 min
cut
off
ends
cucumber
EOS
LSTM
the
person
cut
the
ends
off
the
cucumber
EOS
Encoder
Decoder

Figure 3.3: Proposed architecture for future activity label and caption prediction. In the top figure, the first two fully connected layers (yellow) incorporate the scene context which use object features as input. The two LSTM layers (green) are used to incorporate the sequential activity context which use motion-based features as inputs. The last two fully connected layers (peach) are used to incorporate inter-activity time context which use the last observed activity features (motion-based) as input. There is a fully connected layer (blue) where all these layers are merged together. The output layer (gray) performs the final prediction, where for each element of the future activity sequence, the first few nodes (green) are used as the logistic regression nodes for label prediction. The last node (blue) of the output layer is used as the regression node for starting time prediction. All of the layers have 256 nodes. In the bottom figure, the predicted label and the scene context are then used as input to the encoder LSTM layers and finally the decoder LSTM layers generate the captions. Here, EOS denotes End of Sentence.

sequence of future unobserved activities. For example, if three sequential activities in a video are 'wash objects', 'peel' and 'cut slices', then there may be two probable future activity sequences: 'screw open', 'take out from spice holder', and 'spice' or 'put in bowl', 'puree' and 'smell' (based on two different training instances). But a bowl present in the scene would increase the possibility of the latter sequence.

Several research works on activity recognition [16, 23, 51, 69, 153, 159, 168] and prediction [3, 13] have shown significant performance improvement by using such context information which are also known as context-aware approaches. Most of the existing works have graphical model based approaches for context incorporation. However, they are not very suitable to handle the context of long-term dependency. As mentioned before, LSTM is a popular choice for sequential context incorporation. LSTM networks are straightforward to fine-tune end-to-end and can handle sequential data of varying lengths. So, we use LSTM to incorporate sequential activity context. However, for including the scene context, there is no need for handling such sequential dependency and fully connected layers can capture this efficiently.

The inter-activity time between different activities depends on their labels. For example, we know from experience that ‘peel’ or ‘cut slices’ takes more time than ‘wash objects’. Thus, by observing the previous activity features we can infer about the inter-activity time (difference between the starting time of the observed activity and the future activity) which can be referred to as **inter-activity time context**.

Network Architecture

Our proposed architecture for jointly predicting the labels and the starting time of a future activity sequence is shown in Fig. 3.3. In this case, the LSTM is used to solve a sequential input, sequential output problem. We use the activity features extracted from three (chosen empirically) previously observed activities as the LSTM input. Increasing the sequence length does not improve the prediction accuracy significantly (see Section 3.4.3 for details). We use a two-layer (chosen empirically) LSTM with 256 memory units in each layer. The input of the two fully connected layers in the first branch are the visual features extracted from the objects present in the scene and there are 256 nodes in each layer. The input of the two fully connected layers in the third branch are the activity features extracted from the last observed activity and have 256 nodes in each layer too.

Finally, the outputs from these three branches are combined together and another fully connected layer is added on top of it. The merging combines the effect of different context attributes. In the output layer, for each future activity in the sequence, the first few (equal to the number of activity classes) nodes are used as the logistic regression nodes for sequential label prediction and the last node of the output layer is used as a regression node for predicting the starting time of the future activity sequence.

Model Training Approach

This training method differs from our previous approach [85] in terms of the training procedure. We use the popular open source deep learning package Keras [17] with TensorFlow [1] in the backend which has ready-to-use implementations of LSTM and fully connected layers. The input sequences for the LSTM are chosen in a sliding window manner with a stride of one for data augmentation. For example, to predict the labels of the future sequence containing $(k+1)^{th}$, $(k+2)^{th}$ and $(k+3)^{th}$ activities, activity features extracted from the k^{th}, $(k-1)^{th}$ and $(k-2)^{th}$ activities are used and for predicting the labels of the future sequence containing $(k+2)^{th}$, $(k+3)^{th}$ and $(k+4)^{th}$ activities, activity features extracted from the $(k+1)^{th}$, k^{th} and $(k-1)^{th}$ activities are used and so on. The two fully connected layers in the first branch use visual object features from the scene as input. Another two fully connected layers in the third branch use activity features extracted from the last observed activity as input. We use ReLU activation function for all the fully connected layers. In the output layer, we use softmax activation function in the logistic regression nodes for predicting the label of each activity in the sequence and ReLU activation function in the regression node for predicting the starting time of the sequence. The parameters of the entire network (both of the LSTM and the fully connected layers) are jointly optimized.

We take the summation of the following two losses to compute the final loss. One is the cross-entropy loss function which is defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{X},\mathbf{Y}) &= -\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbf{1}(y^{(i)}=j) \\ &\quad \times \log p(y^{(i)}=j|\mathbf{x}^{(i)}) \end{aligned} \tag{3.1}$$

Here, $\mathbf{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}$ is the set of input feature vectors (activity features of the last three observed activities and features of the objects present in the last observed portion of the scene) in the training dataset, $\mathbf{Y} = \{y^{(1)}, ..., y^{(n)}\}$ is the corresponding set of labels for those input features, and $j = \{1, ..., c\}$ is the set of class labels. $\mathbf{1}(.)$ is an identity function. $\mathbf{x}^{(i)}$ is the sequential activity features extracted from the previous three activities.

The ReLU activation minimizes the mean squared loss between the ground truth inter-activity time and the predicted inter-activity time which is defined as follows:

$$\mathcal{L}(\mathbf{P},\mathbf{Q}) = \frac{1}{n}\sum_{i=1}^{n}(q^{(i)} - \hat{q}^{(i)})^2 \tag{3.2}$$

Here, $\mathbf{P} = \{\mathbf{p}^{(1)}, ..., \mathbf{p}^{(n)}\}$ is the set of input feature vectors (activity feature of the last observed activity) in the training dataset, and $\mathbf{Q} = \{q^{(1)}, ..., q^{(n)}\}$ is the corresponding set of inter-activity times for those input features. $\hat{q}^{(i)}$ represents the predicted inter-activity time given input $p^{(i)}$ where the ground truth inter-activity time is $q^{(i)}$. The outputs of the training are the labels of the three future activities and the starting time of that activity sequence.

The parameters of the network are jointly optimized by minimizing both of these losses. To optimize the network, we use a stochastic gradient descent with an adaptive sub-gradient method

(Adam) [61] which is popular for its strong theoretical convergence guarantee and impressive history of empirical success. We also tested with Adagrad [28], Adamax [61], Nadam [25] and RMSProp [133] but empirically chose Adam. We use Dropout layer [128] with a probability of 0.2 after each layer to prevent overfitting. The batch size is set to 128. We use a learning rate of 0.001.

3.3.2 Caption Generation for Activity Sequences

Role of Scene Context for Label to Caption Mapping

Motivated by the inspiring performance of sequence-to-sequence models in [132] for machine translation and in [140] for video to text mapping, we use a similar model for label to sentence mapping where both the input $(\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_m})$ and the output $(\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n})$ are sequences of words of variable length for each instance. Since the labels do not contain any object information, it is hard to predict the object in the caption only from the label. For example, it is difficult to map from *wash* to *A person washed carrots*. So, we use the scene context from the observed portion along with the label in the encoder LSTM input for meaningful mapping of objects.

Network Architecture

The input to the encoder LSTM is text e.g., *cut apart cucumber*, *take out egg fridge*, *cut off ends carrot*, etc. corresponding to the predicted labels and scene context. In the captions, verbs are followed by objects. To maintain this order, scene context follows the label in the text input. So, sequence-to-sequence learning via encoder LSTM is important here to incorporate this sequential information efficiently and maintain meaningful structure between subject, verb and objects. We do not provide subject as the text input since the subject is constant (*the person*) throughout the dataset. However, for any other dataset where different subjects exist e.g., *man*, *woman*, *boy*, *girl* etc., our network would take the text input in subject-verb-object order as a natural structure. An encoder-decoder LSTM pair is the best option for maintaining meaningful structure between subject, verb and object to incorporate this information correctly. Both the encoder LSTM and the decoder LSTM have 3 layers with 1000 memory units in each layer.

Model Training Approach

In our case, since the caption is always longer than the combination of label and scene context, n is always bigger than m. We estimate the conditional probability $p(\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n} | \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_m})$ given the input $(\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_m})$. At first, we perform embedding by generating a dictionary using all the words in the input of the training set and then convert these words to one hot vectors according to that dictionary. We use one LSTM layer to encode the label to a fixed-dimensional vector and use another LSTM layer to generate a sentence from that vector.

During encoding, the first LSTM generates a sequence of hidden states $(\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_m})$ given the label and the scene context $(\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_m})$. Then a fixed-dimensional vector $\mathbf{z}$ corresponding to the label is generated by the last hidden state of the LSTM. The decoder LSTM computes the conditional probability of the output sentence given the input label and the scene context as follows:

$$\begin{aligned} p\left(\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n} | \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_m}\right) & \\ &= \prod_{d=1}^{n} p\left(\mathbf{b_d} | \mathbf{z}, \mathbf{b_1}, \cdots, \mathbf{b_{d-1}}\right) \end{aligned} \tag{3.3}$$

The distribution $p(\mathbf{b_d} | \mathbf{z}, \mathbf{b_1}, \cdots, \mathbf{b_{d-1}})$ is represented by a softmax over all words in the vocabulary.

During training, the log probability of a correct caption (sentence) is maximized given the label and the scene context. Cross-entropy loss function is used in this model. The batch size we use is 1000. Keras [17] with TensorFlow [1] is the library we use for this work.

3.3.3 Test Case Scenario

For predicting the labels of a future activity sequence, the activity features of the last three observed activities are used in the LSTM input, the features of the objects present in the last observed portion of the scene are used as the input of the first fully connected layers and the activity features of the last observed activity are used as the input of another two fully connected layer. Based on the learned sequence to sequence relationship in the training phase, the network predicts the labels of the next three activities. Using these predicted sequence of labels and observed scene context, the most likely captions for the future activity sequence are generated by the encoder-decoder LSTM pair.

3.4 Experiments

We conduct experiments on three challenging datasets: MPII-Cooking Dataset [117], MPII-Cooking 2 Dataset [119], (fine grained indoor activities) and VIRAT Ground Dataset [94] (coarse outdoor activities) to evaluate the performance of our label prediction framework for a future activity sequence. We do not present the starting time prediction performance for a future sequence since it is exactly the same as presented in [85]. To evaluate the performance of our proposed captioning framework, we conduct experiments on the challenging video description dataset TACoS Multi-Level Corpus [116] built on MPII-Cooking 2 [119]. The goal of the experiments is to compare our predictions with ground truth values, as well as to perform an ablation analysis of the methods.

3.4.1 Datasets

MPII-Cooking Dataset. MPII-Cooking Dataset is a fine grained complex activity dataset where the participants interact with different tools, ingredients and containers to complete a recipe. It has 65 different cooking activities recorded from 12 participants. There are 44 videos with a length of more than 8 hours. The dataset contains a total of $5,609$ annotations [117].

MPII-Cooking 2 Dataset. MPII-Cooking 2 Dataset is a fine grained complex activity dataset where the participants interact with different tools, ingredients and containers to complete a recipe. It has 67 different cooking activities recorded from 30 participants. In total there are 273 videos with a length of more than 27 hours [119].

VIRAT Ground Dataset. VIRAT Ground Dataset is a challenging human activity dataset which consists of 11 different activities recorded in natural outdoor scenes with background clutter. There are total 329 videos with a length of around 5 hours [94]. However, we use only 275 of them as some videos have incomplete annotations.

TACoS Multi-Level Corpus. This video description dataset consists of 185 long indoor videos which contains different actors, fine-grained activities, and small objects in daily cooking scenarios. Each video is annotated by multiple turkers. For each video, there are detailed description with at most 15 sentences, a short description (3-5 sentences), and a single sentence. Since, workers could describe videos without aligning each sentence to the video, the descriptions are natural and have a complex sentence structure [116].

Detailed description of these datasets are available in the supplementary material. These datasets are untrimmed unlike the trimmed datasets popularly used for recognition tasks in activity analysis and have context information. Since we are captioning unobserved future activities, we need untrimmed datasets containing natural sequences of activities with annotated video descriptions. Because of these requirements, the choice of datasets on which our method can be demonstrated is limited. For example, we cannot use MPII-Cooking Dataset [117] or VIRAT Ground Dataset [94] used in [85] since they do not have human descriptions and we cannot use YouCookII Dataset [166] as it does not have the labels annotated in the current version. We cannot use Activity Net Captions [65] because there are only 1.5 activity instances on average in each video which is not enough to incorporate the sequential context for label prediction.

3.4.2 Features

We use C3D (Convolutional 3D) features [135] as activity features for all of the datasets. However, we claim that our method is independent of any particular choice of feature. This is shown in Section 3.4.3 where using bag-of-word based Motion Boundary Histograms (MBH) [20] features gives similar label prediction result for MPII-Cooking Dataset. According to [149], these features are extracted around densely sampled points and a codebook is generated using k-means clustering for these 4000 words long features. Scene context features naturally exist in all of the three datasets.

3.4.3 Label Prediction Results for Activity Sequences

Objective. The main objective of these experiments is to analyze how well our framework can predict the labels of a future unobserved activity sequence.

Performance Measures. The evaluation metrics we use are: 1. multi-class precision (Pr), 2. multi-class recall (Rc), and 3. overall accuracy for top-1 match, top-2 matches and top-3 matches. For all these metrics, the higher value indicates better prediction performance.

MPII-Cooking Dataset [117]	Goal	Precision	Recall	Accuracy % (Top-1)	Accuracy % (Top-2)	Accuracy % (Top-3)
CNN + LSTM [92]	Recognition	34.8	51.7	-	-	-
Proposed Method	Prediction	72.1	67.6	79.9	89.5	93
MPII-Cooking 2 Dataset [119]	Goal	Precision	Recall	Accuracy % (Top-1)	Accuracy % (Top-2)	Accuracy % (Top-3)
Dense trajectories + Hand Trajectories [119]	Recognition	52.2	-	-	-	-
Proposed Method	Prediction	58.8	53.3	65.5	77.4	82.4
VIRAT Ground Dataset [94]	Goal	Precision	Recall	Accuracy % (Top-1)	Accuracy % (Top-2)	Accuracy % (Top-3)
Sparse Autoencoder [42]	Recognition	-	-	54.2	-	-
Graphical Model [13]	Prediction	-	-	68.5	-	-
Proposed Method	Prediction	49.6	22.2	71.8	79.8	86.4

Table 3.1: Label prediction performance comparisons for all of the datasets.

Compared Methods. There is no existing method for predicting future activity labels for MPII-Cooking Dataset and MPII-Cooking 2 Dataset. Therefore, for MPII-Cooking Dataset, we compare with a recent recognition approach [92] which estimates the labels of the *observed* activities. We show that the precision of our prediction of future *unobserved* activities, is higher than that of the recognition method using a combination of CNN and LSTM [92]. For MPII-Cooking 2 Dataset, we compare with a recognition approach [119] which estimates the labels of the *observed* activities and show that precision we achieve for prediction, is higher than that of the recognition method using a combination of dense trajectories and hand trajectories [119]. For VIRAT Ground Dataset, there is an existing graphical model based approach [13] and a semantic scene labeling based approach [63]. We compare our method with [13] and achieve higher accuracy for label prediction. We cannot compare with [63] because they use scene specific customized set of labels which are not annotated in the original dataset. We also compare with a state-of-the-art active learning based recognition approach which uses sparse autoencoder [42] and achieve higher accuracy.

MPII-Cooking Dataset	Accuracy %	Accuracy %
	Next-to-Next Activity	Next-to-Next-to-Next activity
Proposed Method	79.1	78.1
Multi-step Prediction	78.1	77.5

MPII-Cooking 2 Dataset	Accuracy %	Accuracy %
	Next-to-Next Activity	Next-to-Next-to-Next activity
Proposed Method	64.4	63.5
Multi-step Prediction	63.7	62.6

VIRAT Ground Dataset	Accuracy %	Accuracy %
	Next-to-Next Activity	Next-to-Next-to-Next activity
Proposed Method	71.5	69.2
Multi-step Prediction	70.7	68.5

Table 3.2: Sequence prediction performance comparisons for all of the datasets.

To evaluate our label prediction results for further activities in the future sequence, we compare with our previous multi-step prediction baseline [85] where we predicted the next-to-next activity i.e., 2-step prediction (using activity features from the $(i-3)^{th}$, $(i-2)^{th}$ and $(i-1)^{th}$ activities, we predicted the label of the $(i+1)^{th}$ activity) and the next-to-next-to-next activity i.e., 3-step prediction. Multi-step prediction is different from sequence prediction. In multi-step prediction, each prediction step is treated as uncorrelated with the others, while in sequence prediction, the correlations are accounted for.

Experimental Setup. For experiments on MPII-Cooking Dataset, we use five fold leave-one-person-out cross validation approach. Among 12 subjects, we use 7 for training and 5 for testing. For each of the five training instances, we use 7 training subjects and 4 testing subjects for training, leaving 1 from that set for testing. This is done 5 times leaving 1 testing subject out and then averaging the results known as ”five-fold leave-one-person-out” cross validation. For experiments on MPII-Cooking 2 Dataset, we use the experimental setup (same train-test split) of [119]. For experiments on VIRAT Ground Dataset, we use the first 170 videos for training and the rest of them for testing. The network is trained on a NVIDIA Tesla K80 GPU.

Results for MPII-Cooking Dataset. Comparisons of our label prediction results on MPII-Cooking Dataset with the state-of-the-art method are shown in Table 3.1. The method we compare to did not report all of the evaluation metrics we use - hence the missing values. It is seen that our method outperforms the recognition method proposed in [92]. This is not surprising because in recognition problems the network has to decide among all the activity classes whereas in the sequence learning based prediction task, the network needs to consider only a subset of classes which occurred in the training phase after that particular sequence. We achieve similar label prediction accuracy of 79.9% and 80.7% for MPII-Cooking Dataset using C3D and MBH features respectively which justifies the claim that our method is independent of choice of features. Sequence prediction result comparisons with the baseline multi-step prediction method for MPII-Cooking Dataset are shown in Table 3.2. As the prediction horizon increases, there is a gradual accumulation of error. The results show that as the prediction horizon increases, our label prediction accuracy decreases at a slower rate than that of the baseline method. This is intuitive because instead of learning the label of one future activity at a time, the network is learning a sequence of future activity labels now, so, it can infer better about the label of the 2^{nd} or 3^{rd} activity of the sequence than it used to do earlier because of having more information. It is to be noted that even for sequence prediction, prediction results for the first activity in the future sequence have higher accuracy than that of the next activities as we are still using **scene context** from the last observed portion of the scene which is related to the immediate future activity label.

Results for MPII-Cooking 2 **Dataset.** Comparisons of our label prediction results on MPII-Cooking 2 Dataset with the state-of-the-art method are shown in Table 3.1. Our method outperforms the recognition method proposed in [119]. The intuition behind prediction precision being higher than recognition precision is explained above. Sequence prediction result comparisons with the baseline multi-step prediction method for MPII-Cooking 2 Dataset are shown in Table 3.2. For this dataset also, the results show that as the prediction horizon increases, our label prediction accuracy decreases at a slower rate than that of the baseline method.

Results for VIRAT Ground Dataset. Comparison of our label prediction results on VIRAT Ground Dataset with the state-of-the-art methods is shown in Table 3.1. It is seen that our method outperforms the prediction method proposed in [13]. We also achieve higher accuracy than the recognition method proposed by [42]. The intuition behind prediction accuracy being higher than recognition accuracy is explained above. Comparison of the sequence prediction results with the multi-step prediction method for VIRAT Ground Dataset are shown in Table 3.2. Here also, as the prediction horizon increases, our label prediction accuracy decreases at a slower rate than that of the baseline method.

Multiple Possibilities for Future Activity Labels

One particular activity sequence can have multiple possible outcomes. For example, 'wash objects' and 'peel' can be followed by either 'cut apart' and 'cut slices'. As the network has been trained on both of these possible sequences (in one case the network has probably seen 'cut apart' as the next activity and in another case 'cut slices' as the next activity), it is hard to say precisely which is the next activity. Earlier we mentioned that in case of multiple possibilities, such as while choosing between 'spice' or 'put in bowl' after 'wash objects', 'peel' and 'cut slices', a bowl in the scene increases the probability of the activity label being the latter one. But in these types of closely related activities ('cut apart' and 'cut slices'), scene context cannot contribute much as both of the activities require a knife. This is why we present the top-3 choices with the associated probabilities for each of them. We did not go beyond top-3 because after that the probabilities become much lower as we found empirically. In spite of having many closely related ambiguous activities (‘cut dice’, ‘cut slices’, ‘cut apart’) in the dataset, our top-1 match outperforms the baseline in terms of accuracy. Our method can also handle the case of predicting an ‘unknown’ label (never seen in training) when the probability of none of the predicted future activities crosses a threshold.

Effect of Different Context Attributes.

We perform an ablation study to justify the choice of our network. Using only sequential activity context and scene context (eliminating inter-activity time context), we get relatively lower label prediction accuracy for all of the datasets than that of our proposed network. Similarly, using only sequential activity context and inter-activity time context (eliminating scene context), we get lower label prediction accuracy than that of our proposed network for all of the datasets as shown in shown in Table 3.3.

Dataset	Top-1 Accuracy%		
	Proposed Network	Removing inter-activity time context	Removing scene context
MPII-Cooking	79.9	75.7	33.7
MPII-Cooking 2	65.5	60.2	45.7
VIRAT Ground	71.8	69.2	61.0

Table 3.3: Ablation study for label prediction for all of the datasets.

Analysis of Observation Horizon

Here, we will justify the choice of our observation horizon. We empirically chose a sequence length of 3 for preceding activity features as sequence length of 2, 5, 7 and 9 give relatively lower accuracy for MPII-Cooking Dataset as shown in Table 3.4.

Top-1 Accuracy %				
Sequence Length 2	Sequence Length 3	Sequence Length 5	Sequence Length 7	Sequence Length 9
78.6	79.9	79.1	77.5	76.9

Table 3.4: Sequence length sensitivity analysis for MPII-Cooking Dataset.

3.4.4 Captioning Results for Activity Sequences

Objective. The objective of these experiments is to evaluate the quality of the captions generated by our framework against the ground truth captions annotated by the human annotators. More results are presented in the supplementary material.

Performance Measure. The evaluation metrics we use are BLEU (Bilingual Evaluation Understudy) [98], CIDEr (Consensus-based Image Description Evaluation) [138] and METEOR (Metric for Evaluation of Translation with Explicit ORdering) [4]. BLEU is a weighted average of variable length phrase matches against the reference translations in machine translation. CIDEr evaluates how well a candidate sentence matches the consensus of a set of image descriptions. METEOR uses the generalized concept of unigram matching between the machine produced translation and human-produced reference translations. In our case, the number of word matches is compared between the generated captions and the reference captions annotated by the descriptors. For all of the metrics, higher value indicates better performance.

Comparisons. To the best of our knowledge, there is no existing method for generating captions for future unobserved events in videos. Therefore, we compare with [116] which first predicts a semantic representation (SR) of the *observed* portion and then generates detailed captions. We compare against their per sentence BLEU@4 score for short descriptions. We also compare with the BLEU@4, CIDEr and METEOR scores reported in a recent paper [163] which exploits hierarchical

Method	BLEU@4	CIDEr	METEOR
SR Based [116]	22.5	-	-
Hierarchical RNN [163]	30.5	1.602	0.287
Proposed Method	39.2	1.493	0.302

Table 3.5: Comparisons of BLEU@4 (in percent), CIDEr and METEOR scores per sentence for short descriptions in TACoS Multi-Level Corpus. Please note that the SR based method and hierarchical RNN based method report these scores for observed events whereas we report these scores for unobserved future events.

RNNs to generate captions for the observed portion.

Similar to label prediction, since none of the existing methods perform sequence prediction for captions, we can only compare our captioning result for the first unobserved activity with different state-of-the-art methods. However, to evaluate our captioning results for further activities in the future sequence, we compare with multi-step prediction baseline where we predict the next-to-next caption i.e., 2-step caption prediction and the next-to-next-to-next caption i.e., 3-step caption prediction. Multi-step captioning yields different results than sequence captioning because of the same reason as in label prediction.

MPII-Cooking 2 Dataset	BLEU@4	CIDEr	METEOR
Proposed Method (Next-to-Next Caption)	30.2	0.588	0.291
Multi-step Captioning (Next-to-Next Caption)	29.9	0.560	0.274
Proposed Method (Next-to-Next-to-Next Caption)	20.6	0.557	0.264
Multi-step Captioning (Next-to-Next-to-Next Caption)	19.8	0.548	0.254

Table 3.6: Sequence captioning performance comparisons for MPII-Cooking 2 Dataset.

Experimental Setup. For experiments on TACoS Multi-Level Corpus, we use the experimental setup (same test split) of [118] which has also been used in [116]. This information is provided with MPII-Cooking 2 Dataset [119]. We train our network on a NVIDIA Tesla K80 GPU.

Quantitative Evaluation. Comparisons of our video caption generation results on TACoS Multi-Level Corpus with the state-of-the-art methods are shown in Table 3.5. We show that the BLEU@4 score we achieve for the *unobserved* events, is higher than the BLEU@4 score reported in [116]. Our BLEU@4 and METEOR scores are higher than those reported in [163] and our CIDEr score is comparable to the CIDEr score reported in [163] for the *observed* events. Not all of the metrics are reported in [116] - hence the missing values. Quantitative comparisons for captioning a future sequence with the baseline multi-step caption prediction method for MPII-Cooking 2 Dataset are shown in Table 3.6. As the prediction horizon increases, there is a gradual accumulation of error. The results show that as the prediction horizon increases, our captioning performance decreases at a slower rate than that of the baseline method.

Qualitative Evaluation. Qualitative Comparisons for captioning a future sequence with the baseline multi-step caption prediction method for MPII-Cooking 2 Dataset are shown in Table 3.7. Fig. 3.4 depicts an example sequence showing both of our label prediction and captioning results.

No. of Steps	Generated Captions (Multi-Step)	Generated Captions (Proposed Method)	Reference Captions
2	The person **sliced** the leek	The person peeled the leek	The person peeled the leek
3	The person **took out egg**	The person peeled **egg**	The person peeled the leek

Table 3.7: Qualitative comparisons of the generated erroneous captions for multi-step caption generation vs proposed sequential captioning. Mistakes in the captions are marked in bold. Please note that the more we try to predict ahead, the more erroneous the generated captions become.

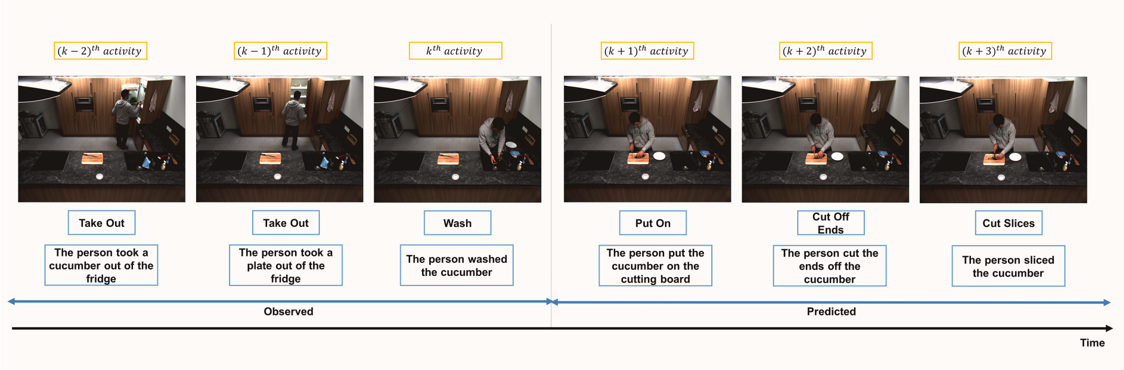

(k − 2)th activity
(k − 1)th activity
kth activity
(k + 1)th activity
(k + 2)th activity
(k + 3)th activity
Take Out
The person took a cucumber out of the fridge
Take Out
The person took a plate out of the fridge
Wash
The person washed the cucumber
Put On
The person put the cucumber on the cutting board
Cut Off Ends
The person cut the ends off the cucumber
Cut Slices
The person sliced the cucumber
Observed
Predicted
Time

Figure 3.4: An example activity sequence showing our label prediction and captioning results on TACoS Multi-Level Corpus.

Effect of the Performance of Label Prediction

We show the BLEU@4, CIDEr and METEOR scores of our generated captions when generated from the ground truth activity labels and when generated from the predicted labels in Table 3.8. The corresponding qualitative comparison for erroneous results are shown in Table 3.9. The type of mistakes made in the generated captions with predicted labels is mostly related to wrong verbs. This is expected since the information regarding the verbs comes from the labels. We get a label prediction accuracy of 65.5% with precision 58.8 and recall 53.3 for MPII-Cooking 2 Dataset [119] which gives an idea about its effect on the evaluation metrics in Table 3.8 obtained using ground truth labels and predicted labels.

Labels Used	BLEU@4	CIDEr	METEOR
Ground Truth Labels	44.0	1.615	0.351
Predicted Labels	39.2	1.493	0.302

Table 3.8: Comparisons of BLEU@4 (in percent), CIDEr and METEOR scores per sentence for short descriptions using ground truth labels vs. predicted labels for caption generation in TACoS Multi-Level Corpus.

Human Description	Generated Captions with Predicted Labels	Generated Captions with Ground Truth Labels
1. The person sliced the carrot 2. The person chopped the herbs	The person **peeled** the carrot The person **cut** the herbs	The person sliced the carrot The person chopped the herbs

Table 3.9: Qualitative comparisons of generated erroneous captions using predicted labels vs. ground truth labels for caption generation in TACoS Multi-Level Corpus. Mistakes in the captions are marked in bold.

Scene Context Used	BLEU@4	CIDEr	METEOR
Ground Truth Scene Context	39.2	1.493	0.302
Predicted Scene Context	30.8	1.033	0.292

Table 3.10: Comparisons of BLEU@4 (in percent), CIDEr and METEOR scores per sentence for short descriptions using ground truth scene context vs. predicted scene context for caption generation in TACoS Multi-Level Corpus.

Effect of Scene Context

MPII-Cooking 2 Dataset [119] has many small objects with similar shapes and appearances. Detecting and recognizing these small objects (sometimes with occlusion) in complex videos is a difficult problem itself. The performance of the object recognition method is crucial to the quality of the generated captions. The error of the object recognition method is propagated in two steps: first during label prediction using predicted scene context and then during the mapping from predicted scene context to objects in the captions.

Using the predicted scene context obtained by the object recognition method used in [119] (combining dense trajectories, hand trajectories and hand cSift features), we compute the BLEU@4, CIDEr and METEOR scores for TACoS Multi-Level Corpus. We show the evaluation metrics of our generated captions when generated from the ground truth scene context and when generated from the

Human Description	Generated Captions with Predicted Scene Context	Generated Captions with Ground Truth Scene Context
1. The person cut an orange in half 2. The person took a plum out of the refrigerator	The person cut the **lime** in half The person took a **onion** out of the refrigerator	The person cut **the** orange in half The person took **plums** out of the refrigerator

Table 3.11: Qualitative comparisons of generated captions using predicted scene context vs. using ground truth scene context for caption generation in TACoS Multi-Level Corpus. Mistakes in the captions are marked in bold.

Observed Sequence Length	2	3	5	7	9
BLEU@4	22.0	39.2	24.0	31.9	38.5
CIDEr	1.117	1.493	1.078	1.142	1.156
METEOR	0.257	0.302	0.262	0.284	0.297

Table 3.12: Comparisons of BLEU@4 (in percent), CIDEr and METEOR scores per sentence for short descriptions using different length of observed activity sequences in TACoS Multi-Level Corpus.

Obs. Seq. Length	Predicted Labels	Generated Captions	Reference Captions
2	cut apart	The person **cut apart** the leek	The person peeled the leek
3	peel	The person peeled the leek	The person peeled the leek
5	slice	The person **sliced** the leek	The person peeled the leek
7	peel	The person peeled the leek	The person peeled the leek
9	slice	The person **sliced** the leek	The person peeled the leek

Table 3.13: Qualitative comparisons of the generated erroneous captions using different length of observed activity sequences in TACoS Multi-Level Corpus. Mistakes in the captions are marked in bold. Please note that in most of these erroneous examples, the verbs are incorrect as a result of incorrectly predicted labels.

predicted scene context using the above mentioned object recognition method in Table 3.10. Please note that the evaluation metrics using our caption generation method with predicted scene context is higher than that of the compared methods as well. The corresponding qualitative comparison for erroneous results are shown in Table 3.11. The type of mistakes made in the generated captions with predicted scene context is mostly related to wrong objects. This is expected since the information regarding the objects comes from the scene context. The mean AP using the above mentioned object recognition method for MPII-Cooking 2 Dataset [119] is 43.7% which gives an idea about the relation between the performance of the object recognition method and the performance of label prediction. This in turn shows the effect of scene context on the performance of caption generation. A better object recognition method will lead to better captioning performance.

Analysis of Observation Horizon

We empirically find that a sequence length of 3 for preceding activity features provides best accuracy for label prediction in MPII-Cooking 2 Dataset [119]. While working with TACoS Multi-Level Corpus, we use observed sequence lengths of 2, 3, 5, 7 and 9 and achieved the highest BLEU@4, CIDEr and METEOR scores for caption generation in TACoS Multi-Level Corpus [116] with an observed sequence length of 3 as shown in Table 3.12. The corresponding qualitative analysis is given in Table 3.13. Although the number of wrong words in each sentence is similar, there is reasonable difference in the values of the evaluation metrics in Table 3.12. This is because the label prediction performance changes as we change the observed sequence length and this in turn changes the number of such erroneously generated captions.

3.5 Conclusions

In this work, we proposed a novel framework for predicting the labels and captions of a sequence of future unobserved activities. We took advantage of the combination of LSTM and fully connected layers to exploit the contextual relationship among activities and objects for label prediction. For mapping the predicted labels and scene context to meaningful captions, we incorporated a sequence-to-sequence based learning approach using an encoder-decoder LSTM pair. Rigorous experimental analysis on challenging datasets proves the robustness of our framework. In future, we plan to extend our prediction method for multi-camera environment and investigate how to predict new unseen activity classes.

Chapter 4

Multi-Sensor Multi-Modal Frame Reconstruction with Conditional GAN

Abstract

Multi-sensor frame reconstruction is an important problem particularly when multiple frames are missing and past and future frames within the camera are far apart from the missing ones. Realistic coherent frames can still be reconstructed using corresponding frames from other overlapping cameras. We propose an adversarial approach to learn the spatio-temporal representation of the missing frame using conditional Generative Adversarial Network (cGAN). The conditional input to each cGAN is the preceding or following frames within the camera or the corresponding frames in other overlapping cameras, all of which are merged together using a weighted average. In autonomous navigation, frame reconstruction is crucial in applications like pedestrian detection, lane detection, SLAM, path planning/navigation etc. Fusion-based approaches incorporating information

from multi-modal data (camera data and mobile terrestrial LIDAR data) are computationally expensive and faster processing is possible if scenes can be reconstructed from LIDAR data only without using any camera data. We propose a cGAN architecture for generating photo-realistic RGB images from 3D point cloud by learning a mapping between these two sensors (camera and LIDAR) or two modalities (image and 3D point cloud). Experiments on three challenging datasets demonstrate that our framework produces comparable results with the state-of-the-art reconstruction method in a single camera and achieves promising performance in multi-camera scenario. Experiments on another challenging dataset demonstrate that our framework achieves promising performance in generating photo-realistic RGB images from 3D point clouds.

4.1 Introduction

Looking at a video sequence with one or more missing frames, how do we infer about what happened in the missing portion? We have never visualized that missing frame. Instead we have a knowledge of the spatio-temporal context of the video to reason about a potential unknown scenario. This spatio-temporal context from the adjacent frames within the camera and the corresponding frames from other overlapping cameras is key to solving an important problem in automated video analysis- frame reconstruction - which is the task of reconstructing missing frames in videos. Frame reconstruction is critical in applications like retrieving missing frames in surveillance videos, anomaly detection, data compression, video editing, video post-processing, animation, spoofing and so on. Although there have been works on frame reconstruction in a single camera setting [15, 53, 131] to the best of our knowledge, ours is the first work to solve it in a multi-camera scenario.

Robust understanding of the environment is vital for ensuring safety and efficiency in autonomous navigation. Autonomous vehicles collect information from the environment using sensors such as monocular camera, LIDAR, stereo binocular camera etc. Monocular cameras capture rich semantic information through high resolution RGB images. But their performance is highly affected by lighting and weather conditions. 3d point clouds provided by mobile terrestrial LIDARs are not very sensitive to these environmental factors and provide distance information as well. However, because of the sparse nature of the data, point clouds cannot represent rich semantic information. Binocular cameras do not perform well in terms of precision and sensor calibration is a prerequisite to use the data obtained from such cameras. A solution to these problems is using fusion-based approaches [105, 106, 122, 127] which combine the advantages of mobile terrestrial LIDAR data and camera data. However, these approaches are computationally expensive and subject to high processing time [97]. In this work, we propose an adversarial approach to learn a mapping between 3D point clouds to RGB images.

Overview of Our Approach. For multi-sensor frame reconstruction, we present an adversarial approach to learn a joint spatio-temporal representation of the missing frame in a multi-camera scenario. First, we learn the possible representations of the missing frame conditioned on the preceding and following frames within the camera as well as on the corresponding frames in other overlapping cameras using conditional Generative Adversarial Network (cGAN) [89] similar to the one used in [52]. Then all of these representations are merged together using a weighted average where the weights are chosen as follows: representations learned from frames within the camera are given more weight when they are close to the missing frame and representations learned from frames in other overlapping cameras are given more weight when the available intra-camera frames are far

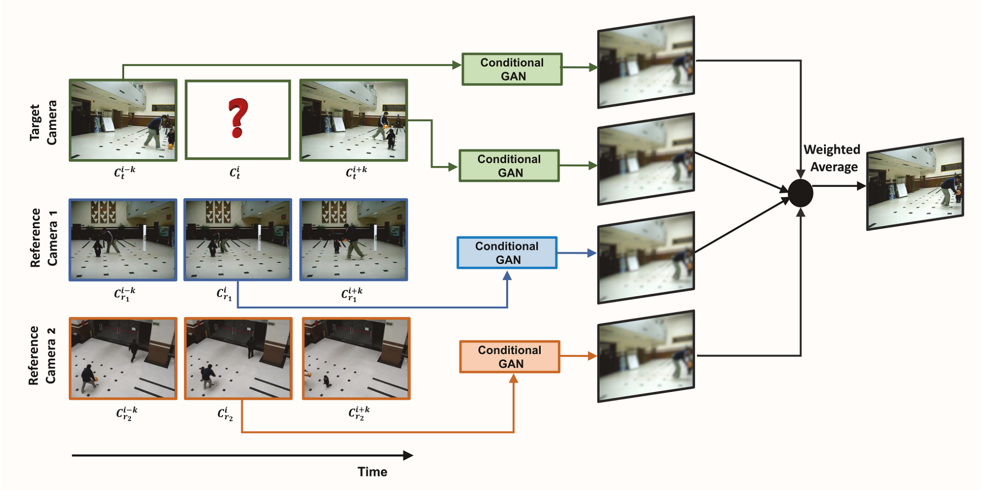


Figure 4.1: An example case of multi-sensor frame reconstruction when there are 3 cameras and the i^{th} frame, C_t^i is missing from target camera 1 of Office Lobby Dataset [35]. We want to generate the missing frame using four available frames (i^{th} frames from reference camera 2 and 3, $C_{r_2}^i$ and C_{r3}^i respectively, and $(i-k)^{th}$ and $(i+k)^{th}$ frames from target camera 1, C_t^{i-k} and C_t^{i+k} respectively). Here, k can be any arbitrary number.

apart. Overview of our proposed framework for multi-sensor frame reconstruction is illustrated in Fig. 4.1.

For multi-modal frame reconstruction, we first generate a depth map from the 3D point cloud and upsample it using a bilateral filtering approach [105] to overcome the limitation associated with the sparse nature of the data. Then we train a cGAN where the conditional input to the cGAN is the upsampled dense depth map and the output is the corresponding RGB image. Overview of our proposed framework for multi-sensor multi-modal frame reconstruction is illustrated in Fig. 4.2. The **main contributions** of our work are:

1. We perform extensive experiments on a challenging multi-camera video dataset to show the effectiveness of our multi-sensor frame reconstruction method.

2. We perform extensive experiments on a single-camera video dataset to provide quantitative

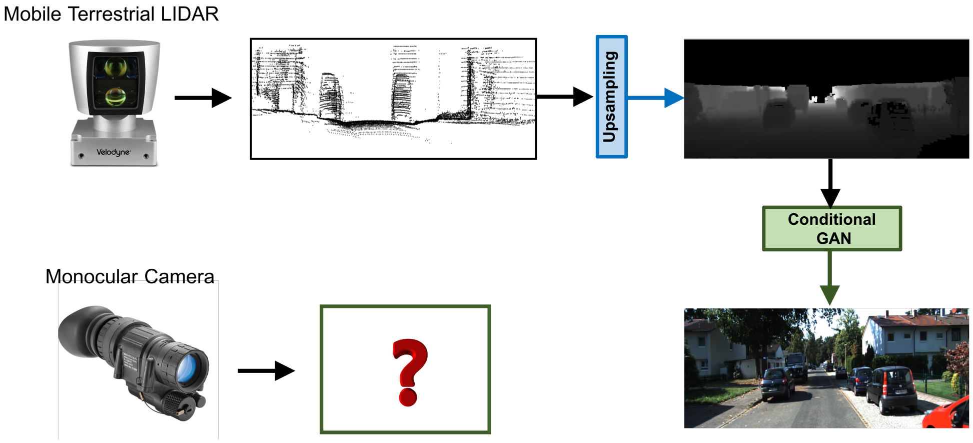


Figure 4.2: An example case of multi-modal frame reconstruction where we want to generate the RGB image using corresponding point cloud from the LIDAR using upsampling and conditional GAN.

comparison of our proposed method with others in the literature.

3. We perform extensive experiments on a challenging autonomous vehicle benchmark dataset to show the effectiveness of our multi-modal frame reconstruction approach.

4.2 Related Works

Our multi-sensor frame reconstruction work is related to video inpainting, frame interpolation, video prediction, frame reconstruction, and generative adversarial networks. There are important differences between frame reconstruction and the problems of video inpainting or frame interpolation. Some spatial information is available in inpainting since the missing portions are assumed to be localized to small spatio-temporal regions. Interpolation cannot reconstruct multiple missing frames as it requires the adjacent (maximum 0.05 seconds apart [131]) frames as inputs. In video prediction, the goal is to predict the most probable future frames from a sequence of past observations.

There are patch-based approaches [91], probabilistic model based approaches [29] and methods handling background and foreground separately [50, 99] for *video inpainting*. For *frame interpolation*, there are approaches [14] using dense optical flow field, phase-based method [87], deep learning approaches [79, 93, 167] and works on long term interpolation [15, 53]. There are sequence-to-sequence learning-based approaches [111, 129], predictive coding network [81], convolutional LSTM [143], deep regression network [146] for *video prediction*. The recent state-of-the-art work on *frame reconstruction* within a single camera [131] uses an LSTM-based interpolation network. However, to the best of our knowledge, there is no work performing frame reconstruction in a multi-camera scenario. This is important when adjacent available frames within the camera are far apart and frames from other corresponding overlapping views can be useful. Recently, *Generative Adversarial Networks* [38] have become popular to solve challenging computer vision problems like text-to-image synthesis [113], frame interpolation [136] and so on. [52] has shown outstanding performance in conditional transfer of pixel-level knowledge. In this work, we seek to leverage GANs for the multi-camera reconstruction problem.

For the multi-modal frame reconstruction problem, to the best of our knowledge, there is one more work [96] which learns a mapping between 3D point clouds from mobile terrestrial LIDARs and RGB images using an adversarial approach without presenting any quantitative results.

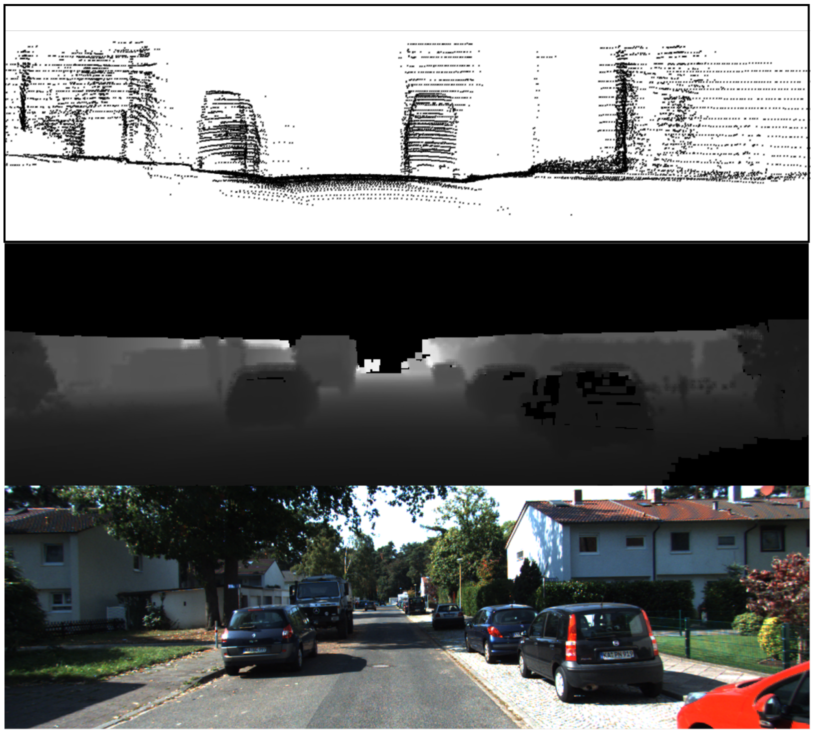


Figure 4.3: An example raw 3D point cloud (top), corresponding upsampled gray image (middle) and ground truth RGB image (bottom).

4.3 Methodology

4.3.1 Data Preprocessing

For multi-sensor frame reconstruction, we resize the images from all the cameras to 256×256 pixels so that they fit into the input of the cGAN. For multi-modal frame reconstruction, some additional processing is required since the 3D point cloud is too sparse to capture meaningful semantic information from the environment. We first create a depth map from the 3D point cloud. Then we upsample it using the modified bilateral filtering approach proposed in [105]. The resultant dense depth map I (output image) is computed as follows [105]:

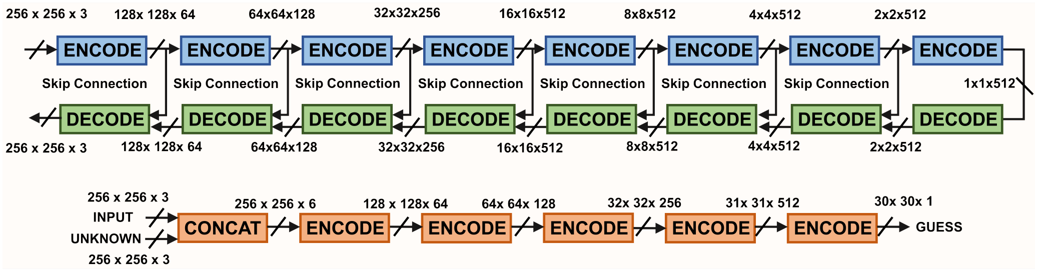


Figure 4.4: Proposed architecture for the generator (top) and the discriminator (bottom) [52]. The pixel values in the 30 × 30 output show how realistic that section of the unknown image is.

$$I_m = \frac{1}{W_m} \sum_{n \in \phi} G_s(\|m - n\|) G_r(|D_m - D_n|) D_n \tag{4.1}$$

Here, D is the sparse depth map and I is the dense depth image. ϕ is the neighborhood mask, I_m is the intensity value of I at pixel position m, and W_m is a normalization factor. G_s weights points at position n inversely to their distance from position m (to decrease the influence of distant pixels), and G_r decreases the influence of points at position n when their intensity values differ from D_m [105].

The upsampling method is similar to convolving the input with a spatial kernel where the kernel size is fixed but the number of points depend on the sparsity of the 3D point cloud [105]. The resultant upsampled depth image shown in Fig. 4.3 (middle) has a black area on the top which is out of the range of the LIDAR. This dense depth map is resized to 256 × 256 pixels and used as the input of the cGAN.

4.3.2 Overall Framework

Similar to general GAN, conditional GAN has a generator and a discriminator. Both of our generator and discriminator have the same architectures used in [109]. We use the conditional GAN to do a mapping between inter-camera or intra-camera frames and between LIDAR point clouds and RGB images. They share an underlying structure i.e., some common low-level information which we want to transfer across the network. Previous image translation problems used an encoder-decoder network [47] where the input was downsampled after being passed through a number of layers and then upsampled using a reverse process when a bottleneck layer was reached [52]. We use a "U-Net"-based architecture of the generator adding skip connection between each layer to overcome the bottleneck problem as the skip connections directly connect encoder layers to decoder layers. L1 loss efficiently captures the low frequency components of images. But using only L1 loss in the objective function for image mapping generates blurry results. We are using a combination of L1 loss and adversarial loss in the objective function. So we aim to use a discriminator efficient in modeling the high frequency components of images. We use the PatchGAN [52] to focus on the structure at local image patches. The discriminator tries to differentiate between the generated and the actual missing frames at patch-level and runs convolutationally across the image to generate an averaged output. So, in this way, the image is modeled as a Markov random field assuming that the pixels separated by more than one patch diameter are independent. The high level network architectures for the generator and discriminator are shown in Fig. 4.4.

4.3.3 Model Training and Inference

In conditional GANs, a mapping is learned from an observed conditional input x and random noise vector z, to an output image y, $G : x, z \rightarrow y$ where the generator G learns to generate outputs close to real images indistinguishable by the discriminator D [52]. The discriminator D learns to efficiently detect the fake outputs generated by G. The objective function of the conditional

GAN is as follows:

$$G^* = E_{x,y}[\log D(x,y)] + E_{x,z}[\log(1 - D(x, G(x,z))] + \lambda E_{x,y,z}[\|y - G(x,z)\|_1] \tag{4.2}$$

Here, $E_{x,y,z}[\|y - G(x,z)\|_1]$ is the $L1$ loss to reduce blurring.

We would refer the camera with the missing frames as the target camera and other cameras as the reference cameras for the multi-sensor frame reconstruction task. Let us assume that there are n overlapping cameras available in a multi-camera scenario. The i^{th} frame, C_t^i, is missing in the target camera. First, we generate two representations of the missing frame from the past and future frame within the camera using two separate conditional GANs. We generate $(\hat{C}_t^i|C_t^{i-k})$ using the past $(i-k)^{th}$ frame and $(\hat{C}_t^i|C_t^{i+k})$ using the future $(i+k)^{th}$ frame. In our case, k can be any arbitrary number based on availability. We generate different representations of the missing frame from the corresponding frame in other reference cameras i.e., generate $(\hat{C}_t^i|C_{r_j}^i)$ where $j = 1 \ldots n$. Basically the network learns a mapping from the observed frames (C_t^{i-k}, C_t^{i+k}, and $C_{r_j}^i$) to the missing frame C_t^i. In accordance with (4.2), C_t^{i-k}, C_t^{i+k}, and $C_{r_j}^i$ are x and C_t^i is y. A training instance is shown in Fig. 4.5.

For the multi-modal frame reconstruction task, the network learns a mapping from the upsampled depth image I^i (from the mobile terrestrial LIDAR) to the corresponding RGB image C^i (from the camera). In accordance with (4.2), I^i is x and C^i is y. A training instance is shown in Fig. 4.6. The generated frame tries to resemble the real frame in terms of the $L1$ loss along with fooling the discriminator. Following [38], we alternate between a gradient descent step upon D and one upon G. Also, in accordance with [38], the training maximizes $\log D(x, G(x,z))$. We divide the objective function in (4.2) by 2 during optimizing D to slow down it learning rate relative to G. To

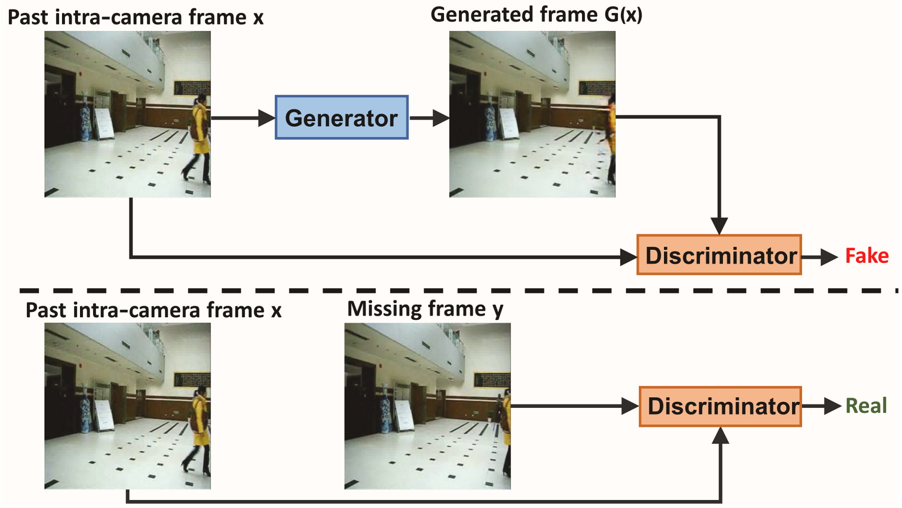


Figure 4.5: A training instance of the conditional GAN for Office Lobby dataset where the discriminator learns to classify between generated and real frames and the generator learns to fool the discriminator.

optimize the network, we use a minibatch stochastic gradient descent with an adaptive sub-gradient method (Adam) [61] and a learning rate of 0.0002.

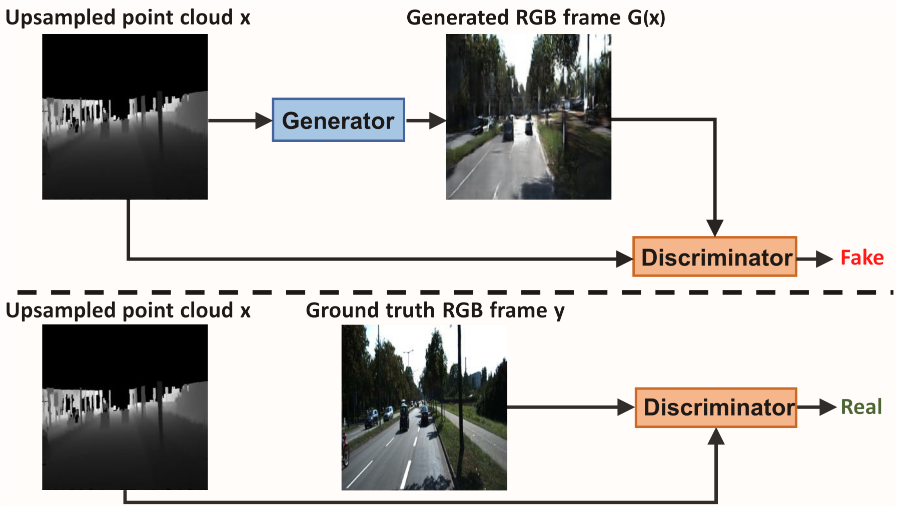


Figure 4.6: A training instance of the conditional GAN for KITTI dataset where the discriminator learns to classify between generated and real RGB images and the generator learns to fool the discriminator.

During testing for the multi-sensor frame reconstruction task, we merge all the generated frames using a weighted average. The weights are chosen by maximizing the average PSNR on a smaller validation set. The more adjacent the available frames are in the target camera, the more weight is given to the representations learned from them than those from the reference cameras. Please note that, since the cameras are partially overlapped, we incorporate the multi-view representation only when there is a person/object present in the overlapping zone.

4.4 Experiments

4.4.1 Datasets

Office Lobby Dataset. Office Lobby Dataset is a multi-camera summarization dataset where 3 video clips are captured by 3 cameras [35]. The cameras are not completely overlapping and the videos have different brightness levels across multi-views. The approximate offset between camera 1 and 2 is about $4.1s$ and between camera 1 and 3 is about $1.33s$. To make an approximate synchronization of the inter-camera frames, these offset values were taken into account while extracting and aligning the frames from different cameras.

Campus Dataset. Campus Dataset is a multi-camera summarization dataset where 4 video clips are captured by 4 cameras [35]. The cameras are not completely overlapping and the videos have different brightness levels across multi-views. The videos are not synchronized and an approximate synchronization of the inter-camera frames were performed to align the frames from different cameras.

KTH Human Action Dataset. KTH Human Action Dataset consists of 6 types of human activities (boxing, handclapping, handwaving, jogging, running, and walking). These actions are performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors with lighting variation [123].

KITTI Dataset. KITTI Dataset is an autonomous vehicle benchmark dataset [36]. We use the 'object' subset which has 7481 training pairs and 7518 testing pairs of camera images and LIDAR point clouds. Both of the sensors are synchronized at $10Hz$. For camera data, we use the RGB images captured by the left camera. Each image has a resolution of 375×1242 pixels. 3D point clouds are collected using a Velodyne HDL-64E 3D laser scanner.

4.4.2 Results

Multi-Sensor Frame Reconstruction

Objective. The main objective of these experiments is to evaluate the quality of the reconstructed frames in multi-camera scenario. We show how the overlapping cameras become more and more important as the distance is increased between the intra-camera frames and the missing frame.

Performance Measure. The evaluation metrics we use are PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index). SSIM estimates how structurally close the reconstructed frame is to the original one. For both of these metrics, higher value indicates better performance. There is no existing work on multi-sensor frame reconstruction to compare our method with. To show the effectiveness of our method in a single camera scenario, we compare with a state-of-the-art reconstruction method [131].

Experimental Setup. We use the standard $80 : 20$ split for training and testing and use TensorFlow [1] to train our network on a NVIDIA Tesla K80 GPU.

Quantitative Evaluation. Our reconstruction results on Office Lobby Dataset and Campus dataset increasing the distance between the missing frame and the available intra-camera past and future frames (multiple frames missing) are shown in Table 4.1 and Table 4.2 respectively. We consider different lengths (gap) of missing frame while testing which are selected in a sliding window manner. Comparisons of our reconstruction results on KTH Human Action Dataset are shown in Table 4.3. We achieve comparable PSNR and SSIM with those reported in [131].

Gap (frames)	1	3	5	7	15	30
PSNR	32.06	29.28	28.10	27.19	25.56	25.17
SSIM	0.95	0.92	0.91	0.90	0.88	0.87

Table 4.1: Multi-Sensor Reconstruction Performance for Office Lobby Dataset.

Gap (frames)	1	3	5	7	15	30
PSNR	34.23	30.57	29.36	28.08	25.11	22.98
SSIM	0.98	0.96	0.95	0.94	0.91	0.89

Table 4.2: Multi-Sensor Reconstruction Performance for Campus Dataset.

Method	PSNR	SSIM
Proposed Method	35.03	0.93
LSTM-Based Method [131]	35.40	0.96

Table 4.3: Single-view Reconstruction Performance Comparisons for KTH Human Action Dataset.

Qualitative Evaluation. Some example results with the conditional input frames and the ground truth missing frames for Office Lobby dataset are shown in Fig. 4.7. Some example results for

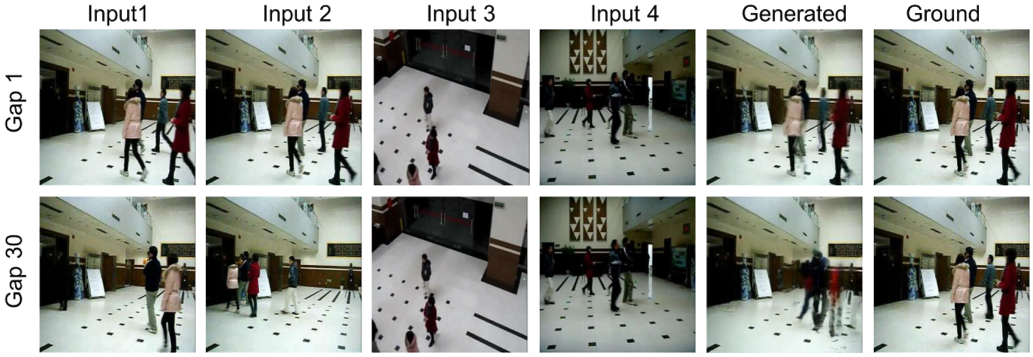


Figure 4.7: Two example results from Office Lobby Dataset where Input 1, Input 2, Input 3, and Input 4 are the preceding and the following frames of camera 1, and the corresponding frames of camera 2 and 3 respectively. As we increase the gap between the preceding and following frames with the missing frame, frames of camera 2 and camera 3 become more important. For example, due to the large number of missing frames in gap 30, the women in red dress is not visible yet in input 1 and her position is far away in input 2. Still, a person wearing a red dress is visible in the correct position of the generated frame incorporating information from the other two cameras.

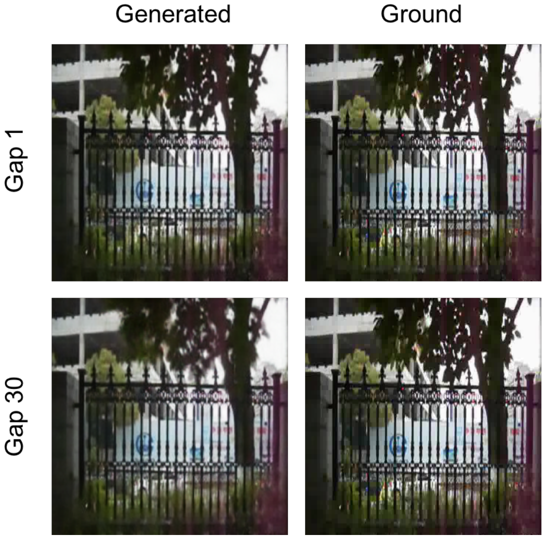


Figure 4.8: Two examples results from Campus Dataset. As expected, the reconstruction performance is better for gap 1 than gap 30.

Campus dataset are shown in Fig. 4.8.

Ablation Study. The comparison of achieved PSNR using only the intra-camera view of camera 1 vs. using multi-sensor reconstruction in Office Lobby Dataset is shown in Table 4.4 as ablation study which justifies the integration of views from multiple sensors specially when the gap is large between the missing frame and the available intra-camera frames.

Gap (frames)	1	3	5	7	15	30
Single	32.06	29.24	28.02	27.02	24.17	23.97
Multi	32.06	29.28	28.10	27.19	25.56	25.17

Table 4.4: Ablation Study for Frame Reconstruction in Office Lobby Dataset

Multi-Modal Frame Reconstruction

Objective. The main objective of these experiments is to evaluate the quality of the reconstructed RGB frames from 3D point clouds.

Performance Measure. Similar to the metrics used in the multi-sensor reconstruction task, we use PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) to evaluate the multi-modal frame reconstruction performance . For both of these metrics, higher value indicates better performance. We cannot compare our results with [96] since they do not provide any quantitative analysis.

Experimental Setup. We use the ‘object’ subset from the KITTI dataset. For training, we use 7481 training pairs of camera images and LIDAR point clouds and for testing we use 7518 testing pairs as provided in the dataset. The network is trained on a NVIDIA Tesla K80 GPU.

Quantitative Evaluation. We achieve a PSNR value of 10.31 and an SSIM value of 0.21 on the test set of KITTI ‘object’ subset.

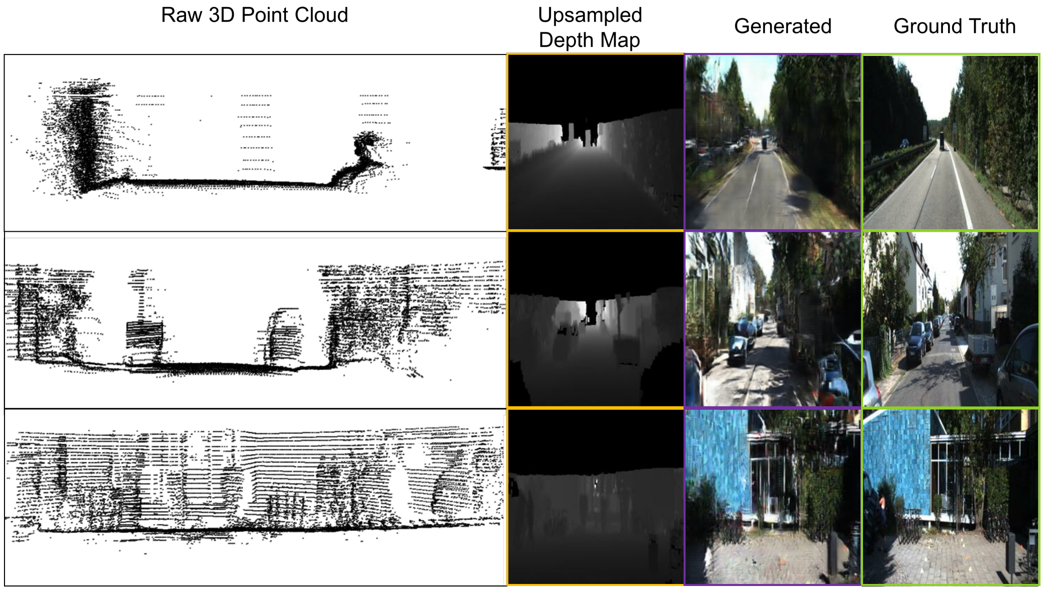


Figure 4.9: Some example results from Kitti Dataset.

Qualitative Evaluation. Some example reconstructed RGB frames, ground truth and corresponding 3D point clouds with upsampling are shown in Fig. 4.9.

4.5 Conclusions

In this work, we proposed an adversarial learning framework for frame reconstruction in multi-camera scenario when one or more frames are missing. We learned the representation of the missing frame conditioned on the past and future frames within that camera as well as the corresponding frames in other overlapping cameras using conditional GAN and merged them together using a weighted average. We used the conditional GAN for another important application which is multi-modal frame reconstruction, where we learn a mapping between 3D point cloud data from mobile terrestrial LIDARs and RGB images from cameras.

Chapter 5

Exploiting Early Prediction for Scalable Video Annotation

Abstract

State-of-the-art video annotation approaches are based on the assumption that the annotator has zero latency for looking up the correct category of label and has to watch the whole video segment. However, in reality getting the correct label from thousands of categories is time consuming and a video segment can be very long. In spite of a lot of interest in this area, two open challenges remain. First, methods need to scale with growing number of video categories. Second, the time spent in watching a video needs to be considered in evaluating the performance of an annotation method. Our proposed method not only reduces the look up time latency, but also minimizes the number of frames required to watch for labeling, hence, the overall annotation cost is reduced. Initially, the most informative queries are selected using label propagation on a similarity graph

and sent to the annotator for annotation. We perform early prediction of the activity labels given the initial frames and dynamically provide suggestions to the annotator in order to reduce the time required for annotation. The annotator provides the correct labels to the queries by taking help from the suggestions without watching the entire video. These newly labeled instances are then used to incrementally update the early prediction model. Our annotation framework achieves comparable recognition performance with the state-of-the-art methods for both ActivityNet1.2, and UCF101 datasets by watching only 7% and 16.9% of the training frames respectively and considering only the top predicted category.

5.1 Introduction

Content-based video classification is a growing field of research due to its various practical applications such as entertainment, multimedia, security, surveillance, etc. Enormous amount of these videos are being generated everyday. Learning a classification model using them requires extensive annotation effort. Data annotation is an expensive task and video data annotation is even more expensive due to the huge number of frames to watch. Moreover, annotation becomes more time consuming due to the higher lookup time of the labels when the number of video categories increases. All these factors contribute to increased video annotation cost, which is a problem for scaling up to large video databases. In this work, we propose a scalable video annotation framework that will reduce the annotation time and cost by a significant margin.

Recent approaches for video annotation [10, 18, 107, 108, 147, 152] overlook the problem of long viewing time of the videos during annotation. When a query video is selected for annotation,

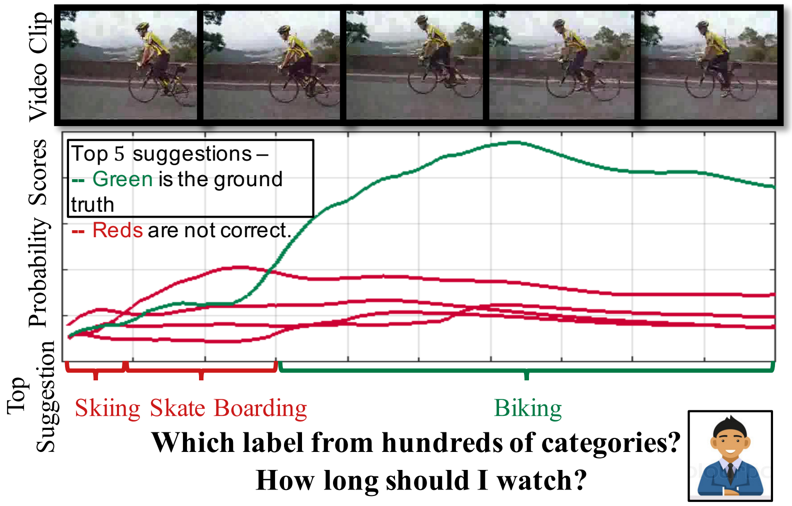


Figure 5.1: The top row shows some frames collected from a video clip that contains a human activity, whereas the bottom row contains a plot of probability scores of activity categories corresponding to that clip. The video segment may belong to one of the hundred categories. It is evident from the plots that after only a few frames the ground truth class is dominant and the ground truth belongs to one of the top 5 suggestions.

it is sent to the annotator assuming that the annotator will provide a label instantaneously irrespective of the length of the video. However, a video can be hundreds or thousands of frames long and the annotation will be expensive if we do not consider this watching time into our problem formulation and performance evaluation. Most of the recent approaches assume that the annotator has to watch the whole video in order to provide the correct label. However, in many cases, few early frames contain distinguishing features which is enough to infer the correct label as shown in Figure 5.1.

Moreover, number of video categories also increases with the growing amount of videos. For example, UCF101 [126], ActivityNet1.2, ActivityNet1.3 [9], Kinetics400 [55], and YouTube-8M [2] have 101, 100, 203, 400, and 4800 activity categories respectively. A video annotation framework has to be scalable in terms of number of activity categories. Given a video to label, an annotator has to lookup a large collection of categories to find the correct label. This process is time

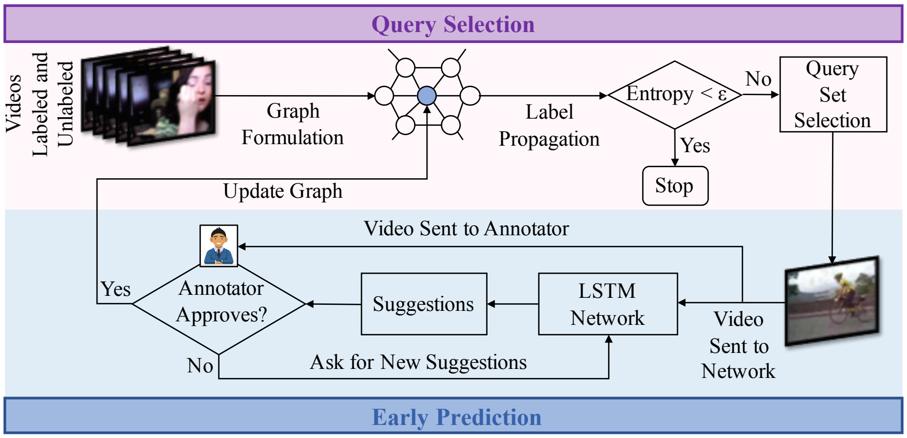


Figure 5.2: Overview of our proposed framework. The framework can be divided into two parts- query selection using a semi-supervised active learning model and early suggestion generation for those queries using an LSTM network. Please refer to Section 5.1.1 and Section 5.3 for details.

consuming and prone to mistakes when the collection is large. It is also impossible for the annotator to memorize every category. Active learning has been proposed recently [26, 43, 46, 77] to reduce the annotation cost. These methods leverage upon the ability of active learning to reduce the number of videos that need to be labeled; however, they do not address the issue of how much time the annotator needs to watch the video for, and the number of categories that he/she needs to consider.

In this work, we aim to solve these two challenges of video annotation. Some previous works [63, 74] performed early label prediction based on few initial frames or in the presence of missing frames. We propose to use an LSTM-based recurrent neural network to continuously predict the labels early after watching few initial frames. The annotator can choose from these labels when he/she is confident about a suggestion, and once he/she does, the annotation is done. Thus, only a small part of the video needs to be watched and the annotator does not have to remember all the labels. We embed our proposed approach within an active learning framework, which minimizes the number of videos that are provided to the annotator, i.e., query selection, in the first place. However, active learning is not the main contribution of this work, and the proposed method could be used with any other query selection approach.

5.1.1 Overview and Main Contributions

A detailed overview of our proposed framework is shown in Figure 5.2. Our goal is to reduce the amount of manual labeling and the time spent in watching the videos during annotation. Given a set of unlabeled and some labeled training instances, we build a graph based on Gaussian similarity measure. We select the most diverse set (the minimum amount required to efficiently train the LSTM network) for initial labeling using a sparse coding based technique [19]. We apply label propagation and transductive inference on this graph to infer on the unlabeled set. Once we perform the label propagation on the graph with few labeled instances, we compute the entropy of the rest of the unlabeled instances. This entropy is the measure of the uncertainty of the current model on the unlabeled set. We select top k highly uncertain training instances as the queries to be labeled by the human annotator. This procedure is performed iteratively until the entropy of the remaining unlabeled data goes below a certain threshold.

Upon receiving these queries, the human annotator starts to watch the long video segment in order to provide the label. We use an LSTM network trained on the labeled training set to generate early suggestions so that the annotator does not have to watch the entire video. The LSTM network has the capability to take the sequential frames as input and produce non-sequential suggestions over time. Based on the output probability distribution of the categories, we show top k categories as the suggestions along with their probability scores. The annotator provides the correct label by taking help from the suggestions when he/she is confident about any of the top k suggestions. These labeled instances are then used to incrementally update the label prediction model so that it can provide better suggestions and similar instances are not selected as queries in future iterations.

Thus, the main contributions of this work can be summarized as follows:

1. We propose a novel approach for reducing video annotation cost by incorporating an early prediction network in an active learning framework. We address the scalability issue for video annotation since our method scales quite efficiently with the number of video categories and significantly reduce both the amount of manual labeling and the long watching time of the videos.
2. We achieve comparable recognition performance with the state-of-the-art methods for both ActivityNet1.2, and UCF101 datasets by watching only 7% and 16.9% of the training frames respectively and considering only the top predicted category.

5.2 Related Works

Video Annotation. Research work in [147] proposes a video annotation framework based on crowdsourcing. It also uses the manually labeled key frames to leverage more sophisticated interpolation strategies to maximize performance under constrained budget. Video annotation method proposed in [108] simultaneously classifies concepts and models correlation between them in order to perform efficient annotation. Research work in [152] proposes a video annotation framework that learns multiple graphs for different important key factors. In [18], they annotate near-scenes sharing the same concept or semantic meaning. [10] uses an active learning framework for temporal action localization. However, these approaches are not scalable with the number of video categories. We address this scalability issue in our proposed framework.

Activity Recognition and Prediction. Visual feature-based activity recognition approaches can be classified into three broad categories such as interest point-based low-level local features, human tracking and pose-based mid-level features, and semantic attribute-based high-level features based methods. The survey article [104] contains more detailed review on feature-based activity recognition. Recently, activity recognition methods have been benefited from the use of deep learning techniques such as convolutional two-stream network [33], R*CNN [37], differential RNN [139], Temporal Segment Network (TSN) [151], Two-Stream Inflated 3D ConvNet (I3D) [12] etc. Research works in [74] and [63] perform early prediction of activity labels, whereas LSTM-based RNN is used for early detection of activities in [83]. Some recognition approaches [70, 160, 168] use context information as well. In [3, 13, 85], unobserved activity labels are predicted without any observation. However, most of the above mentioned methods involve batch-learning algorithms requiring all of the training instances to be present and labeled beforehand. In contrast, we combine early prediction with active learning in order to reduce manual effort for video annotation.

Active Learning. Active learning has been successfully applied to many computer vision problems including tracking [148], object detection [142], image [6] and video segmentation [32], and activity recognition [42, 43, 44, 45, 46, 100]. To the best of our knowledge, none of the active learning methods designed for video annotation takes into account the latency for looking up the correct category of label or the time spent in watching the entire video.

Long Short-Term Memory (LSTM) Network for Video Analysis. LSTMs have been popular to analyze temporal information because of the ability to handle long-term dependency. For video analysis, Donahue et al. take advantage of LSTM-based RNN for visual recognition with large scale labeled data [24]. Du et al. build an RNN in a hierarchical way to recognize actions [27]. [85] and [3] use LSTM-based networks for predicting unobserved activity labels. Here, we exploit an LSTM-based recurrent neural network for generating early suggestions for the annotator.

5.3 Approach

In this section, we discuss about different parts of our proposed framework in details. We start with explaining different components of the active learning framework used for query selection for the sake of completeness. Then we discuss our proposed LSTM-based early prediction network which is incorporated in this framework to reduce manual labeling effort.

5.3.1 Query Selection using Active Learning

This semi-supervised active learning method includes similarity graph formation, label propagation, initial sample selection and entropy-based query selection.

Similarity Graph Formation

We use both of the labeled and the unlabeled videos to construct a graph based on their feature similarity [7, 22]. We rely on this similarity of data to infer about the label of the unlabeled data using only the labeled data. The geometry can be defined by a graph $G = (V, E)$ where the nodes $V = \{1, \ldots, N\}$ represent the activity instances, both labeled and unlabeled, and the edges E represent similarity between them. These similarities are given by a weight matrix $\mathbf{W}$, such that $\mathbf{W_{ij}}$ is non-zero if x_i and x_j are neighbors. We compute the weight matrix using the following Gaussian kernel -

$$\mathbf{W_{ij}} = \exp(-\gamma \|x_i - x_j\|^2) \tag{5.1}$$

Label Propagation

After constructing the graph, some of its nodes are initialized with ground truth labels. The criterion behind choosing these nodes is discussed in the next section. At any time, we have three types of nodes - nodes belonging to the labeled training set, nodes belonging to the unlabeled training set, and nodes belonging to the unlabeled test set. We want to smoothly propagate the learned information and do not want a query to be selected for manual labeling that is similar to the previous queries. For this purpose, we use label propagation which is very efficient in propagating new information and performing more accurate inference.

Given the graph G and some labeled nodes, each node starts to propagate its label to its neighbor and the process is repeated until convergence or maximum allowed iterations. We use a label spreading algorithm similar to Zhou [165]. At each step, a node i receives a contribution from its neighbors j (weighted by the normalized weight of the edge (i, j)), and an additional small contribution given by its initial value. The detailed algorithm [7] is given as follows,

Algorithm 1 Label spreading

```
Compute the affinity matrix $\mathbf{W}$ using Eqn. 5.1
Compute the diagonal degree matrix $\mathbf{D}$, $\mathbf{D}_{ii} = \sum_j W_{ij}$
Compute the graph Laplacian $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$
Initialize $\hat{Y}^{(0)} = (y_1, \ldots, y_l, -1, -1, \ldots, -1)$
Iterate $\hat{Y}^{(t+1)} = \alpha\hat{Y}^{(t)} + (1-\alpha)\hat{Y}^{(0)}$ until convergence
Label point $x_i$ by the sign of $\hat{Y}_i^{(t+1)}$
```

Initial Sample Selection

We leverage sparse coding technique instead of naive approaches like random or serial selection, for selecting the training samples to be initially labeled. This ensures that the graph is initialized with the optimal set of labels and during first few iterations, higher test accuracy can be achieved using fewer manually labeled examples. We select the minimum amount of diverse training instances required to efficiently train the LSTM network [19]. The problem can be formulated as,

$$\begin{aligned} &\min \|X - XZ\|_F^2 \\ &\text{s.t. } \|Z\|_{2,1} \leq \tau, \end{aligned} \tag{5.2}$$

where, $X = \{x_i \in R^d,\ i = 1, \ldots, N\}$. Each x_i represents the feature descriptor of an activity instance. N denotes the number of instances in the batch. $Z \in R^{N \times N}$ is the sparse coefficient matrix and $\|Z\|_{2,1} = \sum_i^N \|z_i\|_2$ is the row sparsity regularizer, i.e., sum of l_2 norms of the rows of Z. τ is the parameter for the level of sparsity. Using Lagrange multipliers, optimization problem in 5.2 can be written as,

$$\min \frac{1}{2}\|X - XZ\|_F^2 + \lambda\|Z\|_{2,1}, \tag{5.3}$$

where, λ is the trade-off parameter. We implement the algorithm using an Alternating Direction Method of Multipliers (ADMM) optimization framework [8].

Entropy-Based Query Selection

Given the set of labeled and unlabeled videos, the goal is to select a subset of the unlabeled videos which are most informative for the current model. Here, we consider entropy or model uncertainty as the measure of informativeness. We then send these videos to the annotator to watch and label. Entropy of an instance x_i is given by

$$h(x_i) = -\sum_{c \in C} p_c \log(p_c), \tag{5.4}$$

where, C is the set of class labels and p_c is the probability of class c. We select a subset S of size k from the unlabeled set $U = \{x_i\}$.

$$\underset{S \subset U \,\wedge\, |S|=k}{\arg\max} \; H(S), \tag{5.5}$$

where, $H(S)$ is the entropy of a set S, which can be computed as follows,

$$H(S) = \sum_{x_i \in S} h(x_i) \tag{5.6}$$

Our problem setting motivates us to use a statistical reasoning known as transductive inference [137] which is capable of utilizing the abundance of unlabeled examples along with the labeled ones. Given the similarity graph, we perform transductive inference on the unlabeled examples and compute entropies. We select some training examples with higher entropy and send them for manual labeling. Once we get the label, we continue this process until the entropy of the system is below a certain threshold.

5.3.2 Early Prediction

In this work, the LSTM network predicts the labels of the query videos from few initial frames and provide suggestions to the annotator. It is shown from the experiments that, using this approach, annotation can be performed with significantly reduced budget since the annotator can decide the label before watching the entire video once he/she is confident about any of the top suggestions made by the early prediction network. At each time step, the LSTM network predicts the label of the current query video using the features from the frames it has seen so far and this can be performed in real time.

Sequential Suggestions

The goal of the early prediction network is to decrease the viewing time as well as the number of possible categories the annotator has to look through. After each iteration of label propagation, the selected queries are sent to both the annotator and the LSTM network. We use the features extracted from a sequence of video frames as the input to the LSTM network and in the output the network produce a probability distribution of the classes as shown in Figure 5.3. For the first iteration, we use an LSTM network trained only on the initially labeled training samples in the similarity graph. After that for each iteration, we dynamically update our LSTM model with more training data as more training samples get labeled by the annotator.

As the annotator starts watching a video, the network starts predicting the label of that video and these prediction scores are generated as a function of time. Over time, the network gets access to more and more features extracted from the increasing number of frames and the predictions become more accurate. The annotator receives these suggestions and can stop watching the video once he decides on one of the top k labels. The network not only helps the annotator to reduce viewing time but also enables him to look through only k possible categories instead of hundreds or thousands of categories. However, there is a trade-off between these two which is analyzed in Section 5.4. This is because if the annotator wants to rely on a prediction made by the network at an earlier stage, he/she might have to look at a higher number of possible candidates to make sure that the annotation is correct. Whereas when the annotator decides to watch the video for a longer time, the prediction scores become much more accurate and a smaller value of k can guarantee that the top k prediction contains the correct label.

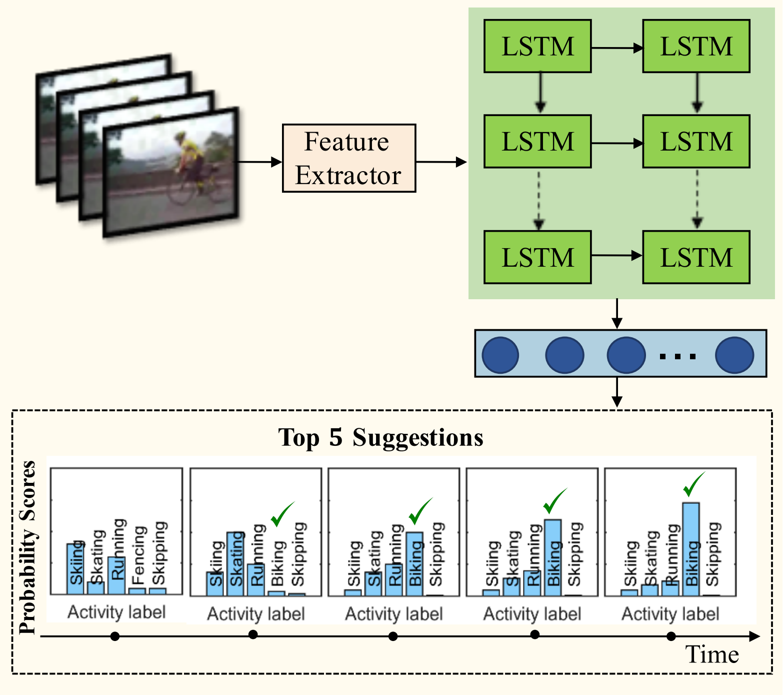


Figure 5.3: Features collected from the video frames are provided as the input to the LSTM network. The network generates prediction of the activity classes at each time stamp. The top k predictions are shown to the annotator as suggestions.

Model Architecture and Training

We choose an LSTM-based early prediction network since LSTMs [48] are suitable to incorporate long-term sequential dependency and do not suffer from the vanishing and exploding gradient problem common in traditional RNNs. In this framework, the LSTM network sequentially processes the incoming video frames and continuously generates top k prediction scores. We empirically find that a two-layer LSTM network with 256 nodes in each layer followed by a Dropout layer [128] with a probability of 0.2 after each layer performs better than any other architectures.

In order to learn the suggestions from the feature sequences, we use I3D features [12] of dimension 2048 as the input to the network as shown in Figure 5.3. Maximum sequence length for an activity segment is T. We either zero pad or cut sequences if they are smaller or bigger than T. We employ a many-to-many sequence learning strategy. That means, if an input activity segment has the representation of size $T \times 2048$, the target label is of size $T \times C$, where, C is the number of classes. In the output layer, we use softmax activation function in the logistic regression nodes and use the cross-entropy loss function which is defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{X}, \mathbf{Y}) &= -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{1}(y^{(i)} = j) \\ &\quad \times \log p(y^{(i)} = j | \mathbf{x}^{(i)}) \end{aligned} \tag{5.7}$$

Here, $\mathbf{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}$ is the set of input feature vectors from the training videos, $\mathbf{Y} = \{y^{(1)}, ..., y^{(n)}\}$ is the corresponding set of labels, and $j = \{1, ..., c\}$ is the set of class labels. $\mathbf{1}(.)$ is an identity function. For a particular training instance, $\mathbf{x}^{(i)}$ represents the activity features extracted from that video. We use RMSProp [133] as the optimizer with an initial learning rate of 0.001 and Keras [17] with Tensorflow [1] back-end for implementing the network.

Test Case Scenario

When a query video is sent to the network, it starts processing the incoming video frames sequentially and generates the top k suggestions. The annotator has to limit either the number of frames he/she wants to watch or the number of suggestions he/she wants to look into. If k is fixed, then the annotator will continue watching until he/she is confident about one of the top k predictions made by the network and if the number of watched frames is fixed then the annotator can decide on a higher value of k to get the correct label. The overall framework is portrayed in Algorithm 2.

Algorithm 2 Overall framework

```
Input: Training activity segments, $A^l = \{a_i^l\}$ and testing activity segments, $A^u = \{a_i^u\}$
Output: Accuracy on $A^u$, the most informative queries, and suggestions generated by LSTM network $\mathcal{L}$
Extract motion and appearance features for the activity segments in $A^l$ and $A^u$.
  We use an off-the-self I3D model.
  It generates $2048$ dim. features for each $16$ frames.
Use above I3D features to train $\mathcal{L}$ (Sec. 5.3.2).
  Use $\mathcal{L}$ to generate top suggestions for the annotator.
Construct a graph, $G = (V, E)$ (Sec. 5.3.1)
  $V$ contains activities from both $A^l$ and $A^u$.
  Use pre-computed features in the nodes.
  Use Gaussian similarity for the edge weights.
Give labels to some of the nodes ($k$) belongs to $A^l$.
  Use sparse coding to select diverse set. (Sec. 5.3.1)
while Entropy $(A^l) > \epsilon$ do
    Run label propagation (Algo. 1) on $G$ to compute -
      Marginal probabilities of the nodes $A^l$ and $A^u$.
      Compute the entropies of the nodes.
    Perform Query selection on G (Sec. 5.3.1)
      Select the most informative set of size $k$.
    Send the $k$ queries to annotator for manual labeling.
      Generate the suggestions from $\mathcal{L}$. (Sec. 5.3.2)
      Send the suggestions along with the queries.
    Give labels to these $k$ nodes belongs to $A^l$ in $G$
end while
Compute and report the accuracy on $A^u$.
Report the amount of manual labeling and effort.
```

5.4 Experiments

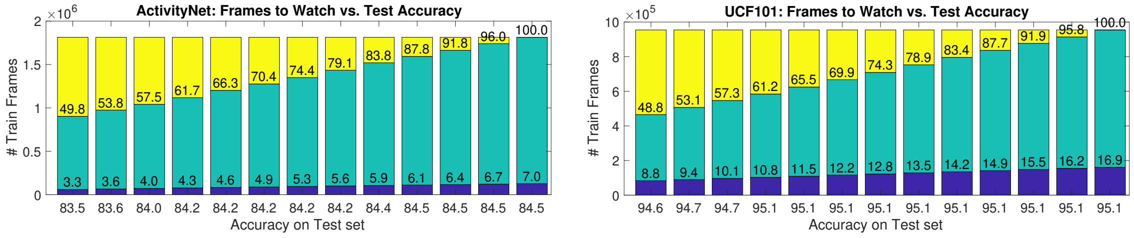
Dataset - ActivityNe1.2: ActivityNet [9] is a large-scale video benchmark for human activity understanding. This dataset has 4819 training videos, 2383 validation videos and 2480 test videos. ActivityNet version 1.2 provides samples from 100 activity classes with an average of 1.5 temporal activity segments per video. These videos were collected from Youtube and have the properties of being "wild."

Dataset - UCF101: UCF101 [126] is an action recognition dataset collected from YouTube, having 101 action categories and 13320 videos. With the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc, it is one of the most challenging datasets. The videos in 101 action categories are grouped into 25 groups, where each group consists of 4-7 videos of an action.

Features: We use I3D features of size 2048 for each sixteen frames extracted from the Kinetics pre-trained I3D network [12].

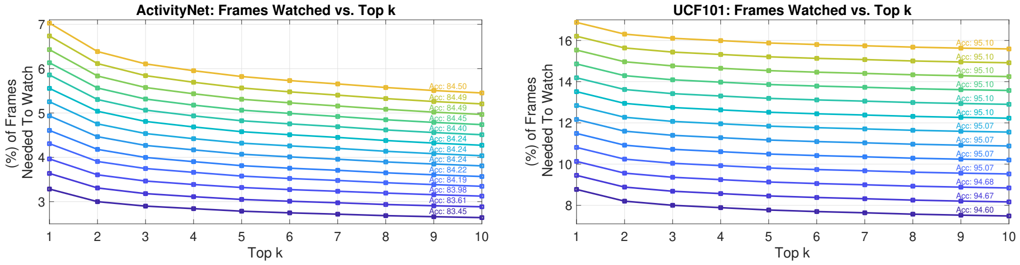
Experiment Setup and Objectives: We use the train-test split provided with the dataset for UCF101. It has 9537 training instances and 3783 testing instances. For ActivityNet, we use 4819 training videos and 2383 validation videos as used in the literature [151]. The network is trained on a NVIDIA Tesla K80 GPU.

We use Scikit-Learn [101] for graph formulation and label propagation. We use a γ value of 0.1 in the label spreading kernel and a τ value of 40 for the level of sparsity. Inference on this graph provides us entropies and predicted labels of all of the unlabeled nodes. Then, we perform active learning on the graph in order to select the nodes for manual labeling based on the entropy as discussed in Section 5.3.1. We preform these operation in an iterative manner until the entropy of the entire set reaches to a minimum level. Starting with an LSTM network trained on the initially labeled training videos, we dynamically update our LSTM model with more training data as more training samples get labeled by the annotator. For an activity segment, the network generates a sequence of probability scores over time. As the annotator watches the video segment, these suggestions pop up for the corresponding frame number. This allows the annotator to pick up the correct label in the shortest possible time. If none of the suggested labels appear to be correct by the annotator, the annotator can provide a new label. Upon sufficient number of such examples, we can incrementally train the model to handle such cases in future. We simulate the decision of the real human annotators using ground truth labels and update the model with new labeled training sample when any of the top k suggestions made by the LSTM matches the ground truth.



(a) (b)

Figure 5.4: The bar charts show the reduction of human annotation effort with respect to the accuracy over test set for ActivityNet and UCF101. (a), and (b) corresponds to ActivityNet and UCF101 respectively where $k = 1$, i.e., the annotator is looking at the top prediced category only. The yellow bar represents the total number of frames in the training set, the green bar represents the percentage of training frames sent to the annotator and the purple bar represents the percentage of training frames the annotator needs to watch for correct annotation. This figure is best viewed in color.

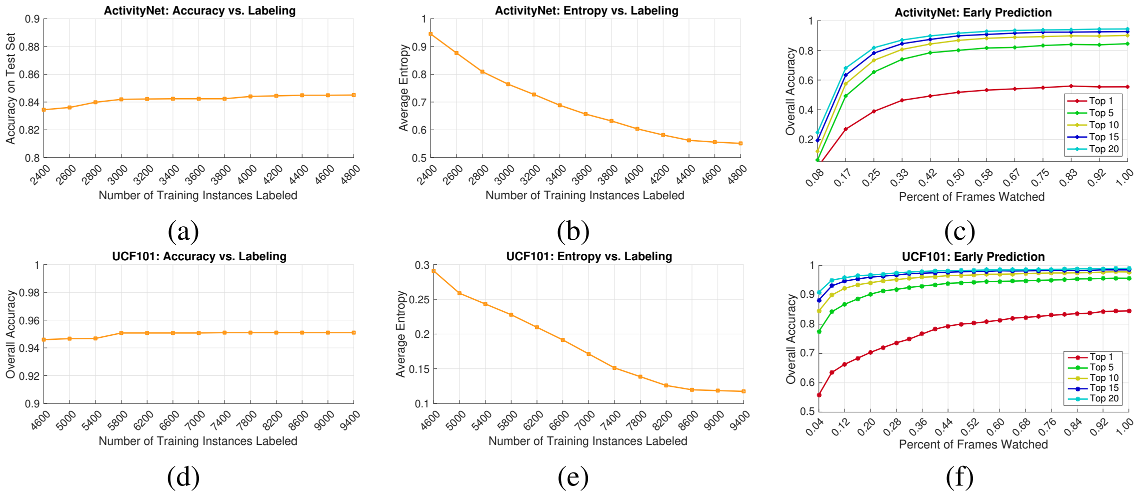


(a) (b)

Figure 5.5: These plots shows the trade-off between the percentage of frames needed to watch and the number of categories the annotator has to look at in each annotation step for ActivityNet and UCF101. This figure is best viewed in color.

We conduct a number of experiments in order to show the effectiveness of the proposed framework for video annotation. Through our experiments, we will show that our framework not only reduces manual effort by a huge margin but also matches state-of-the-art approaches in large-scale activity recognition. The main objectives are as follows,

1. To show how efficient our framework is in reducing the amount of time required for labeling (Figure 5.4) which is the main contribution of the work.
2. To show how the percentage of frames watched and the number of categories the annotator needs to look at vary (Figure 5.5) in each iteration and the trade-off between them.
3. To show how efficient our framework is in terms of recognition performance (Figure 5.6(a), (b), (d), and (e).
4. To show how effective LSTM network is in performing early prediction (Figure 5.6(c) and (f).



(a) (b) (c)

(d) (e) (f)

Figure 5.6: Four of these plots illustrate accuracies (a,d) and average entropies (b,e) of two datasets during label propagation. At each iteration, we find the k most informative instances, provide labels to them, perform label propagation again, and report accuracies and entropies. Plots (c) and (f) illustrate the effectiveness of suggestion generation using the early prediction network. This figure is best viewed in color.

Reduction of Human Effort: One of the main contributions of the proposed framework is its ability to reduce the human effort and thus annotation cost by a great margin in terms of both number of labels and viewing hours. The bar charts in Figure 5.4 illustrate how much cost reduction can be achieved as a function of test set accuracy for ActivityNet and UCF101. For each bar chart, the Y-axis represents the number of frames belonging to the training set, while X-axis represents the accuracy over the test set. The huge gap between the green and the purple bar shows how the early prediction network reduces the viewing time to a great extent maintaining the same test accuracy.

The interpretation of the bars is as follows - for example if we look at the first bar in Figure 5.4(a), 49.8% of the total training frames are sent to the annotator, but he/she has to watch only 3.3% of the training frames and look at the top prediction ($k = 1$) to annotate the next set of queries. This results in 83.5% accuracy on the test set. For the next bar, 200 more data have been sent for annotation which increases both the number of frames sent to the annotator and the number of frames needed to be watched. After this annotation, 83.6% test accuracy is achieved in the next iteration of label propagation. In the last bar, by labeling 100% of the training frames and watching only 7% of them, we can achieve 84.5% accuracy on the test data. This is a huge margin for annotation cost reduction, since the annotators normally charge by hour. Also huge time is saved as the lookup time latency is reduced since the annotator has to look at only the top predicted category instead of the long list of 100 labels for annotation. The bar charts in Figure 5.4(b) illustrate how much cost reduction can be achieved as a function of test set accuracy for UCF101 when looking at the top predicted category.

In Figure 5.5(a) and (b), we show the trade-off between the percentage of frames watched and the number of categories the annotator needs to look at in each iteration for ActivityNet and UCF101 respectively. The Y-axis represents the percentage of frames needed to watch, while X-axis represents the number of categories the annotator has to look at (each curve corresponds to each annotation step with a fixed test accuracy). The annotator has to watch a higher number of frames if he/she chooses to look at fewer number of categories and vice versa. For example, in 5.5(a), at the final annotation step, when all training data are manually labeled and the test accuracy is 84.5%, the annotator has to watch 5.7%, 5.8%, and 6.1% of the training frames when decided to look at top 7, top 5, and top 3 predictions respectively.

Recognition Performance: The plots in Figure 5.6(a) and (d) shows the recognition accuracies over test set. The X-axis represents the amount of labeled instances so far at each iteration, whereas, the Y-axis represents the accuracies. For ActivityNet (Figure 5.6(a)), we have 4819 training and 2383 testing videos. We initially select $k = 2400$ videos belonging to the training set using sparse coding technique as discussed in Section 5.3.1 required to efficiently train the LSTM network. At each iteration, we select additional $k = 200$ videos for manual labeling. It is evident from the plots that as we add more and more labeled data, accuracy increases over time. Our method shows comparable performance with the state-of-the-art method Temporal Segment Network (TSN) [151] (86.3% mAP on the test set) by achieving an accuracy of 84.5% and mAP of 86% on the test set when it uses all of the manually labeled training instances. As we are giving labels to more and more training nodes, the plots become saturated.

For UFC101 (Figure 5.6(d)), we have 9537 training and 3783 testing videos. We initially select $k = 4600$ videos belonging to the training set using sparse coding technique as discussed in Section 5.3.1. At the beginning, only these videos are labeled and rest of them are unlabeled. At each iteration, we select additional $k = 200$ videos for manual labeling based on the method described in Section 5.3.1. Our method shows comparable performance with the state-of-the-art method [12] (98% accuracy on the test set using I3D features) by achieving an accuracy of 95.1% and mAP of 96% on the test set when it uses all of the manually labeled training instances.

Reduction in System Entropy: While the experimental setup remains same, the plots in Figure 5.6(b) and (e) shows the overall reduction of system entropy as we add more and more labeled data for both of the datasets. As expected, the entropy of the system decreases as we add more labeled videos. Please note that these curves monotonically decrease instead of being saturated as the accuracy vs. labeling curves because there is no exact correlation between average entropy and accuracy. Let us consider two separate examples where the final prediction is correct. In one case, the entropy can be lower because the top class probability is very high. For another, even with a higher top class probability the entropy can be higher if the top class probability is close to those of the other classes. In both cases the model is accurate but the entropy is very different.

Early Prediction Performance: We conduct experiments to investigate the effectiveness of our suggestion generator as shown in Figure 5.6(c) and (f) for ActivityNet and UCf101 respectively. X-axis shows the fraction of frames watched, wheres Y-axis show the Top k accuracies, where $k = 1, 5, 10, 15, \text{ and}, 20$. We see that the plots get saturated after a small percentage of the frames have been watched. For ActivityNet, top 10 suggestions are accurate in 80% cases even before watching 33% of the frames. For UCF101 top 10 suggestions are accurate in 95% cases even before watching 25% of the frames. Please not that these plots show results on the test set when the LSTMs are trained on the entire training set.

5.5 Conclusions

In this work, we presented a novel video annotation approach by taking scalability and viewing time into account. We used a semi-supervised active learning technique with an LSTM-based early prediction network. We selected the most informative queries using label propagation and calculated the entropy of the nodes. Then the LSTM-based early prediction network is used for generating label suggestions which help to reduce manual effort significantly. Experimental evaluation shows that our framework reduces the annotation cost by a significant margin.

Chapter 6

Conclusions

6.1 Thesis Summary

Near-future prediction in videos is an active research area in the computer vision community because of its growing importance in real-life application which require anticipatory response. The future can be represented in terms of labels, captions, frames etc. each one having its own strength and weakness. In this thesis, we explore several prediction tasks (i.e., label prediction, starting time prediction, captioning, and multi-sensor multi-modal frame reconstruction) focusing on developing efficient data driven solutions. Since all of these tasks require huge amount of labeled data which is expensive in terms of annotation time and cost, we also explore an efficient solution for scalable video annotation.

In Chapter 2, we presented an LSTM-based deep network leveraged on different context attributes from the observed portion of the video to jointly predict the labels and starting times of future unobserved activities. In Chapter 3, leveraged on our label prediction framework, we presented a sequence-to-sequence learning-based approach using an encoder-decoder LSTM pair for captioning near-future activity sequences. In Chapter 4, we proposed conditional Generative Adversarial Network (cGAN) for multi-sensor multi-modal frame reconstruction. Finally, in Chapter 5, we presented an early prediction framework which can be combined with any active learning framework so that video annotation becomes scalable. Experimental results show that our methods achieve significant performance gain over existing approaches and baselines in standard benchmark datasets.

6.2 Future Research Directions

6.2.1 Prediction for Planning and Navigation Strategy

In Chapter 2, we proposed an LSTM-based deep network for jointly predicting the labels and starting times of future unobserved activities using observed context information. In Chapter 3, we proposed a sequence-to-sequence learning-based approach for captioning near-future activity sequences. It would be interesting to extend our approaches for trajectory prediction in path planning and navigation strategy. One approach could be incorporating complex dynamic models in our existing framework for such purpose. The solution would have meaningful impact in applications like autonomous navigation and active sensing.

6.2.2 Transfer Learning for Generative Models

Transfer learning is widely used for discriminative models using fine-tuning. However, to the best of our knowledge, there has been only one work [154] which focus on transfer learning for generative models. Successful generative models are data-hungry and require huge amount of data for efficient training which is expensive to obtain. These models suffer a significant loss in performance or collapse completely when asked to perform a new task or provided with a new unseen dataset. Transfer learning for generative models can reduce convergence time and improve the quality of generated samples when target data is limited. In Chapter 4, we proposed conditional Generative Adversarial Network for multi-sensor multi-modal frame reconstruction. One interesting extension would be to explore transfer learning approaches for such generative models using pre-trained GANs especially when there is a lack of sufficient training images.

6.2.3 Continual Learning for Generative Models

Another challenging future direction of work is to explore continual lifelong learning approaches for generative models. While learning a new task, neural networks have the tendency to overwrite the parameters necessary to perform well at a previously trained task. This chronic phenomenon where training for a new task catastrophically degrades the system's performance on previously learned tasks is known as catastrophic forgetting [34]. One solution is to replay all former data but this requires large memory and not practical since access to previous data is limited in real life applications. Continual learning facilitates learning from a data distribution that changes with time and thus retains important information. Although there have been a number of works on continual learning for discriminative models based on rehearsal, regularization, activations etc. [57, 62, 76, 78, 80, 112, 115], it has a lot of potential to be explored for generative models [73, 95, 110, 124, 125, 156]. This can be an interesting future direction of our work since generative models have been proven to be effective for solving many popular computer vision problems.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[3] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, pages 5343–5352, 2018.

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[5] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al. Video in sentences out. *arXiv preprint arXiv:1204.2742*, 2012.

[6] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[7] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. *Semi-supervised learning*, 10, 2006.

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[10] Fabian Caba Heilbron, Joon-Young Lee, Hailin Jin, and Bernard Ghanem. What do i annotate next? an empirical study of active learning for action localization. In *ECCV*, pages 199–216, 2018.

[11] S. Cao, K. Chen, and R. Nevatia. Activity recognition and prediction with pose based discriminative patch model. In *WACV*, pages 1–9, 2016.

[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017.

[13] A. Chakraborty and A. Roy-Chowdhury. Context-aware activity forecasting. In *ACCV*, pages 21–36, 2014.

[14] Kanglin Chen and Dirk A Lorenz. Image sequence interpolation using optimal control. *Journal of Mathematical Imaging and Vision*, 41(3):222–238, 2011.

[15] Xiongtao Chen, Wenmin Wang, and Jinzhuo Wang. Long-term video interpolation with bidirectional predictive network. In *IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2017.

[16] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, pages 3273–3280, 2011.

[17] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[18] Chien-Li Chou, Hua-Tsung Chen, Suh-Yin Lee, et al. Multimodal video-to-near-scene annotation. *IEEE TMM*, 19(2):354–366, 2017.

[19] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE TMM*, 14(1):66–75, 2012.

[20] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441, 2006.

[21] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, pages 2634–2641, 2013.

[22] Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. Efficient non-parametric function induction in semi-supervised learning. In *AISTATS*, volume 27, page 100, 2005.

[23] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, M. Roshtkhari, J. Mehrsan, and G. Mori. Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191*, 2015.

[24] J. Donahue, L. A. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[25] T. Dozat. Incorporating Nesterov momentum into adam. Technical report, Stanford University, Tech. Rep., 2015.[Online]. Available: http://cs229. stanford. edu/proj2015/054 report. pdf, 2015.

[26] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. Exploring representativeness and informativeness for active learning. *IEEE transactions on cybernetics*, 47(1):14–26, 2017.

[27] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[28] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[29] Mounira Ebdelli, Olivier Le Meur, and Christine Guillemot. Video inpainting with short-term windows: application to object removal and error concealment. *IEEE Transactions on Image Processing*, 24(10):3034–3047, 2015.

[30] Wafa Elmannai and Khaled Elleithy. Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. *Sensors*, 17(3):565, 2017.

[31] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, pages 15–29, 2010.

[32] Alireza Fathi, Maria Florina Balcan, Xiaofeng Ren, and James M Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011.

[33] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[34] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[35] Yanwei Fu, Yanwen Guo, Yanshu Zhu, Feng Liu, Chuanming Song, and Zhi-Hua Zhou. Multi-view video summarization. *IEEE Transactions on Multimedia*, 12(7):717–729, 2010.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *ICCV*, pages 1080–1088, 2015.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[39] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772, 2014.

[40] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, pages 2712–2719, 2013.

[41] Patrick Hanckmann, Klamer Schutte, and Gertjan J Burghouts. Automated textual descriptions for a wide range of video events with 48 human actions. In *ECCV*, pages 372–380, 2012.

[42] M. Hasan and A. Roy-Chowdhury. Continuous learning of human activity models using deep nets. In *ECCV*, pages 705–720, 2014.

[43] M. Hasan and A. Roy-Chowdhury. Context aware active learning of activity recognition models. In *ICCV*, pages 4543–4551, 2015.

[44] Mahmudul Hasan, Sujoy Paul, Anastasios I Mourikis, and Amit K Roy-Chowdhury. Context-aware query selection for active learning in event recognition. *IEEE TPAMI*, 2018.

[45] Mahmudul Hasan and Amit K Roy-Chowdhury. Incremental activity modeling and recognition in streaming videos. In *CVPR*, pages 796–803, 2014.

[46] Mahmudul Hasan and Amit K Roy-Chowdhury. Incremental learning of human activity models from videos. *CVIU*, 144:24–35, 2016.

[47] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[48] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[49] D. A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, pages 489–504, 2014.

[50] Kuo-Lung Hung and Shih-Che Lai. Exemplar-based video inpainting approach using temporal relationship of consecutive frames. In *IEEE Int. Conf. on Awareness Science and Technology (iCAST)*, pages 373–378. IEEE, 2017.

[51] M. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, pages 1971–1980, 2016.

[52] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[53] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, pages 4565–4574, 2016.

[55] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[56] SR Ke, H. L. U. Thuc, YJ Lee, JN Hwang, JH Yoo, and KH Choi. A review on video-based human activity recognition. *Computers*, 2(2):88–131, 2013.

[57] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[58] Muhammad Usman Ghani Khan and Yoshihiko Gotoh. Describing video contents in natural language. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 27–35, 2012.

[59] Muhammad Usman Ghani Khan, Lei Zhang, and Yoshihiko Gotoh. Human focused video description. In *ICCV Workshop*, pages 1480–1487, 2011.

[60] Muhammad Usman Ghani Khan, Lei Zhang, and Yoshihiko Gotoh. Towards coherent natural language description of video streams. In *ICCV Workshop*, pages 664–671, 2011.

[61] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[62] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[63] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, pages 201–214, 2012.

[64] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2):171–184, 2002.

[65] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. *arXiv preprint arXiv:1705.00754*, 2017.

[66] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, volume 1, page 2, 2013.

[67] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE TPAMI*, 35(12):2891–2903, 2013.

[68] T. Lan, T. C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, pages 689–704, 2014.

[69] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, pages 1216–1224, 2010.

[70] Tian Lan, Weilong Yang, Yang Wang, and Greg Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010.

[71] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[72] Mun Wai Lee, Asaad Hakeem, Niels Haering, and Song-Chun Zhu. Save: A framework for semantic annotation of visual events. In *CVPR Workshop*, pages 1–8, 2008.

[73] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. *arXiv preprint arXiv:1812.09111*, 2018.

[74] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE TPAMI*, 36(8):1644–1657, 2014.

[75] W. Li and M. Fritz. Recognition of ongoing complex activities by sequence prediction over a hierarchical label space. In *WACV*, pages 1–9, 2016.

[76] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[77] Hongsen Liao, Li Chen, Yibo Song, and Hao Ming. Visualization-based active learning for video annotation. *IEEE TMM*, 18(11):2196–2205, 2016.

[78] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE, 2018.

[79] Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Int. Conf. on Computer Vision (ICCV)*, volume 2, 2017.

[80] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[81] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[82] M. Lukasik, P. K. Srijith, T. Cohn, and K. Bontcheva. Modeling tweet arrival times using log-Gaussian cox processes. In *EMNLP*, pages 250–255, 2015.

[83] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, pages 1942–1950, 2016.

[84] T. Mahmud, M. Hasan, A. Chakraborty, and A. Roy-Chowdhury. A Poisson process model for activity forecasting. In *ICIP*, pages 3339–3343, 2016.

[85] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *ICCV*, pages 5773–5782, 2017.

[86] Catharine LR McGhan, Ali Nasir, and Ella M Atkins. Human intent prediction using markov decision processes. *Journal of Aerospace Information Systems*, 12(5):393–397, 2015.

[87] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1418, 2015.

[88] R. Minhas, A. A. Mohammed, and QM J. Wu. Incremental learning in human action recognition based on snippets. *IEEE TCSVT*, 22(11):1529–1541, 2012.

[89] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[90] Brendan Morris, Anup Doshi, and Mohan Trivedi. Lane change intent prediction for driver assistance: On-road design and evaluation. In *Intelligent Vehicles Symposium (IV)*, pages 895–901, 2011.

[91] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[92] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *CVPR*, pages 1020–1028, 2016.

[93] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Int. Conf. on Computer Vision (ICCV)*, pages 261–270, 2017.

[94] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, JK Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, pages 3153–3160, 2011.

[95] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, and Moin Nabi. Learning to remember: Dynamic generative memory for continual learning. 2018.

[96] Zhenchao Ouyang, Yu Liu, Changjie Zhang, and Jianwei Niu. A cgans-based scene reconstruction model using lidar point cloud. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 1107–1114. IEEE, 2017.

[97] Michel Owayjan, Ali Hayek, Hassan Nassrallah, and Mohammad Eldor. Smart assistive navigation system for blind and visually impaired individuals. In *International Conference on Advances in Biomedical Engineering (ICABME)*, pages 162–165, 2015.

[98] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.

[99] Kedar A Patwardhan, Guillermo Sapiro, and Marcelo Bertalmío. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007.

[100] Sujoy Paul, Jawadul H Bappy, and Amit K Roy-Chowdhury. Non-uniform subset selection for active learning in structured data. In *CVPR*, pages 830–839, 2017.

[101] 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.

[102] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.

[103] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[104] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 2010.

[105] C. Premebida, J. Carreira, J. Batista, and U. Nunes. Pedestrian detection combining RGB and dense LIDAR data. In *IROS*, pages 0–1. IEEE, Sep 2014.

[106] Cristiano Premebida and Urbano Nunes. Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection. *The International Journal of Robotics Research*, 32(3):371–384, 2013.

[107] Stanislav Protasov, Adil Mehmood Khan, Konstantin Sozykin, and Muhammad Ahmad. Using deep features for video scene detection and annotation. *Signal, Image and Video Processing*, pages 1–9, 2018.

[108] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *ACM MM*, pages 17–26. ACM, 2007.

[109] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[110] Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. Lifelong generative modeling. *arXiv preprint arXiv:1705.09847*, 2017.

[111] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[112] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[113] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[114] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *ICCV*, pages 3696–3705, 2017.

[115] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[116] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *GCPR*, pages 184–195, 2014.

[117] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, pages 1194–1201, 2012.

[118] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *ICCV*, pages 433–440, 2013.

[119] 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. *IJCV*, 119(3):346–373, 2016.

[120] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, pages 1036–1043, 2011.

[121] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241, 2012.

[122] Joel Schlosser, Christopher K Chow, and Zsolt Kira. Fusing lidar and images for pedestrian detection using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2198–2205. IEEE, 2016.

[123] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *IEEE Int. Conf. on Pattern Recognition*, volume 3, pages 32–36, 2004.

[124] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, 2017.

[125] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[126] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012.

[127] Luciano Spinello and Roland Siegwart. Human detection using multimodal and multidimensional features. In *2008 IEEE International Conference on Robotics and Automation*, pages 3264–3269. IEEE, 2008.

[128] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[129] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Int. Conf. on machine learning*, pages 843–852, 2015.

[130] Chen Sun and Ram Nevatia. Semantic aware video transcription using random forest classifiers. In *ECCV*, pages 772–786, 2014.

[131] Ximeng Sun, Ryan Szeto, and Jason J Corso. A temporally-aware interpolation network for video frame inpainting. *arXiv preprint arXiv:1803.07218*, 2018.

[132] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[133] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. Technical report, 2012.

[134] Eric C Townsend, Erich A Mielke, David Wingate, and Marc D Killpack. Estimating human intent for physical human-robot co-manipulation. *arXiv preprint arXiv:1705.10851*, 2017.

[135] D. Tran, L. Bourdev, R. Fergus L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[136] Joost van Amersfoort, Wenzhe Shi, Alejandro Acosta, Francisco Massa, Johannes Totz, Zehan Wang, and Jose Caballero. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *arXiv preprint arXiv:1711.06045*, 2017.

[137] Vladimir Vapnik. *Statistical learning theory. 1998*, volume 3. Wiley, New York, 1998.

[138] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[139] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *ICCV*, pages 4041–4049, 2015.

[140] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015.

[141] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[142] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1-2):97–114, 2014.

[143] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[144] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.

[145] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *CVPR Workshops*, pages 41–48, 2016.

[146] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, pages 98–106, 2016.

[147] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 101(1):184–204, 2013.

[148] Carl Vondrick and Deva Ramanan. Video annotation and tracking with active learning. In *NIPS*, 2011.

[149] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[150] L. Wang, X. Zhao, Y. Si, L. Cao, and Y. Liu. Context-associative hierarchical memory model for human activity recognition and prediction. *IEEE Transactions on Multimedia*, 2016.

[151] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE TPAMI*, 2018.

[152] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song. Unified video annotation via multigraph learning. *IEEE TCSVT*, 19(5):733–746, 2009.

[153] X. Wang and Q. Ji. Video event recognition with deep hierarchical context model. In *CVPR*, pages 4418–4427, 2015.

[154] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 218–234, 2018.

[155] X. Wei, P. Lucey, S. Vidas, S. Morgan, and S. Sridharan. Forecasting events using an augmented hidden conditional random field. In *ACCV*, pages 569–582, 2014.

[156] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: learning to generate images from new categories without forgetting. *arXiv preprint arXiv:1809.02058*, 2018.

[157] Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. A multi-scale multiple instance video description network. *arXiv preprint arXiv:1505.05914*, 2015.

[158] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, volume 5, page 6, 2015.

[159] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, pages 17–24, 2010.

[160] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[161] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[162] Haonan Yu and Jeffrey Mark Siskind. Learning to describe video with weak supervision by exploiting negative sentential information. In *AAAI*, pages 3855–3863, 2015.

[163] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016.

[164] A. Zammit-Mangion, M. Dewar, V. Kadirkamanathan, and G. Sanguinetti. Point process modelling of the Afghan War Diary. *Proceedings of the National Academy of Sciences*, 109(31):12414–12419, 2012.

[165] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *NIPS*, 16(16):321–328, 2004.

[166] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv:1703.09788*, 2017.

[167] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European Conf. on computer vision*, pages 286–301. Springer, 2016.

[168] Y. Zhu, N. M. Nayak, and A. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *CVPR*, pages 2491–2498, 2013.

[169] Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot. Long short term memory for driver intent prediction. In *Intelligent Vehicles Symposium (IV)*, pages 1484–1489, 2017.