Efficient Annotation of Objects for Video Analysis

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS in Computer Science and Engineering
by
Research

by

Sirnam Swetha
201303014
sirnam.swetha@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " Efficient Annotation of Objects for Video Analysis" by Sirnam Swetha, has been carried out under our supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Vineeth N Balasubramanian
Date	Adviser: Prof. C.V. Jawahar

To my Family

Acknowledgements

As I submit my thesis, I wish to extend my gratitude to all those people who helped me in successfully completing this journey.

Foremost, I would like to express my deepest gratitude towards my advisers Prof. C. V. Jawahar and Dr. Vineeth N Balasubramanian for their intellectual guidance, support, and encouragement at every phase of the project. They have been an incessant source of inspiration from the beginning and I am immensely inspired by their work ethics. I have been fortunate enough to have worked under and learned from Prof. C. V. Jawahar and Dr. Vineeth N Balasubramanian. The learning has not only helped me academically but also helped me become a better person. I would also like to extend my special gratitude towards Dr. Guruprasad Hegde for accepting to work with me.

It was great fun and an unforgettable experience for me to be part of CVIT. I would also like to thank my lab mates at IIIT for all the intellectual discussions and their valuable feedbacks Jay Panda, Anand, Suriya, Mohak, Koustav, Riddhiman, Thrupthi, Aniket, Praveen, Rahul, Arwa, Ajitesh, Nikhil, Priyam, Anurag, Pramod, Rajat, Vidyadhar and many other CVITians. I would also like to thank all the professors, TAs and seniors for their teachings, lessons, and guidance. I thank Pujitha, Nikhil, Sharvani, and Saifulla for being such amazing friends. Special thanks to Rajan, Silar, Siva, Satya and Varun for all of their help on various occasions.

Last but definitely above all, I would like to thank my family for their unconditional love and support. I could not have accomplished it without the support and understanding of my mother, brother, grandparents and my family. I shall be forever indebted to my uncle for the financial support, faith in me and helping me in pushing my boundaries. I would like to specially acknowledge my brother, who has always played a very encouraging and supporting role in all my endeavours. I would like to wholeheartedly thank my family for their continuous encouragement and persistent support throughout my years of study.

At the end, I would like to thank God, for everything.

Abstract

"How dangerous it always is to reason from insufficient data."

SHERLOCK HOLMES

The field of computer vision is rapidly expanding and has significantly more processing power and memory today, than in previous decades. Video has become one of the most popular visual media for communication and entertainment. In particular, automatic analysis and understanding the content of a video is one of the long-standing goals of computer vision. One of the fundamental problems is to model the appearance and behavior of the objects in videos. Such models mainly depend on the problem definition. Typically, in many scenarios, the change in problem statement is followed by the changes in the annotation and its complexities. Creating large-scale datasets in this scenario using the manual annotation process is monotonous, time-consuming and non-scalable. In order to address this challenge and strive towards practical large scale annotated video datasets, we investigate methods to autonomously learn and adapt object models using temporal information in videos.

Even though the vision community has advanced in field of problem solving but data generation and annotation is still a tough problem. Data annotation is expensive, tedious and involves a lot of human efforts. Even after data annotation, it is essential to validate the goodness of annotations, which again is a tiresome process. To address this problem, we investigate methods to autonomously learn and adapt the object models using temporal information in videos. This involves learning robust representations of the video. The aim of this thesis is two-fold, first we propose solutions for efficient and accurate object annotation mechanisms in video sequences and secondly, to raise awareness in the community about the importance and attention it deserves.

As our first contribution, we propose an efficient, scalable and accurate object bounding box annotation method for large scale complex video datasets. We focus on minimizing the annotation efforts simultaneously increasing the annotation propagation accuracy to get a precise and tight bounding box around object of interest. Using a self training approach, we propose a combination of semi-automatic initialization method with an energy minimization framework to propagate the annotations. Using an energy minimization system for segmentation gives accurate and tight bounding boxes around the object. We have quantitatively and qualitatively validated the results on publicly available datasets.

In the second half, we propose annotation scheme for human pose in video sequences. The proposed model is based on a fully-automatic initialization, from any generic state-of-the-art method. But the initialization is prone to error due to the challenges in video data type. We exploit the availability of redundant information from the redundant data type. The model is build on the temporal smoothness assumption in videos. We formulate the problem as a sequence-to-sequence learning problem, the architecture uses Long Short Term Memory encoder-decoder model to encode the temporal context and annotate the pose. We show results on state-of-the-art datasets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "It is a capital mistake to theorize before one has data."
>
> SHERLOCK HOLMES

It is an undisputed agreement that large amounts of annotated training data is one of the crucial reasons for the success of deep learning. They have been the primal reason for the considerable progress in the field, not just as source of large amounts of training data, but also as means of measuring and comparing performance of competing algorithms. Imagenet [28] dataset with 1 Million annotated samples is an example that depicts the reason behind one of the very first successfull attempts for object classification using neural networks. Since then there have been many advancements in the representation learning but surprisingly there are no such advancements to create datasets with bigger sizes.

Researchers have shown impressive results on tasks such as large scale object detection, object recognition, image classification, pose estimation, action recognition, segmentation and event detection [4, 23, 24, 28, 34, 58, 73, 74, 79, 83]. However, these methods are far from mature when it comes to deploying in practical applications. This can be ascribed to the following reasons. First, generating large scale training data with wide variety of anticipated scenarios and annotating them. Apart from training data, another barrier is generating a large amount of validation and test datasets that can help to find the model complexity and benchmark the algorithms. For practical applications, one must test a solution rigorously for hundreds of hours to determine the performance robustly. Hence, it is crucial to annotate large scale datasets. Datasets are an integral part of research and they have been the chief reason for the considerable progress in the field, not just as source of large amounts of training data, but also as means of measuring and comparing performance of competing algorithms. In the past, people have attempted to generate annotated data in computer vision (eg. ImageNet, crowd-sourcing).

In the process of generating large annotated datasets, many opt for manual execution, where an annotator marks the Ground Truth (GT) annotation in a set of images [35, 43, 66]. More recently researchers have started to use video sequences to generate GT as they provide much richer representations of the object [1, 56]. Manual annotation especially in videos involves a huge cognition load, and is subject to

inefficiency and inaccuracies [93]. This is more evident while annotating humans as the limbs might move in a nonlinear manner and is difficult to capture the variation in shape and extent of the object within neighboring frames. The above aspects can be addressed either by crowd-sourcing or by developing semi automatic annotation schemes using computer vision and machine learning techniques. In this work, we propose annotation schemes that use vision and machine learning algorithms to annotate objects in video sequences.

1.1 Problem Definition

In this work, we focus on efficient object bounding box annotation in videos and accurate human pose annotation in videos.

Object Annotation in Videos. Object Annotation is one of the most fundamental problems in vision community because of its application in a wide variety of tasks and it is also a part of many high-level problems. We propose an efficient and yet accurate annotation scheme with tight bounding boxes, for object in videos with minimal supervision The annotations are propagated across the frames using self-learning based approach. An energy minimization scheme for the segmentation is the core component of our method. Figure 1.1 shows an example of annotation of a person walking on road. Some of the applications of object annotation are motion analysis, event detection, surveillance systems, transport, sports analytics.

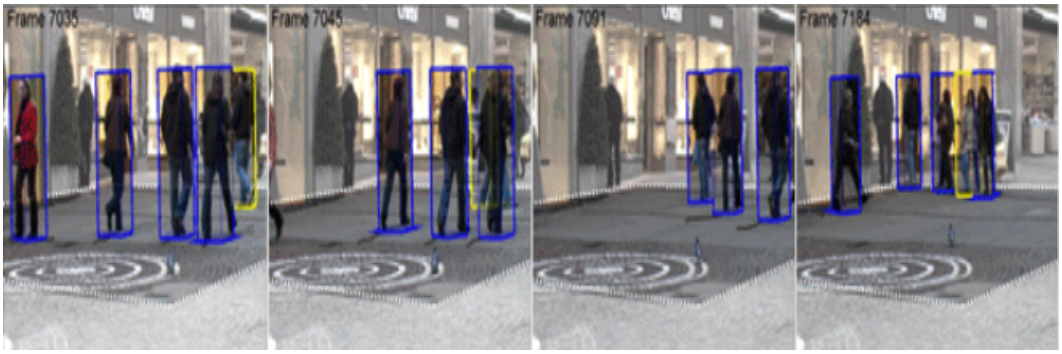


Figure 1.1: Example showing objects (human) being annotated with a tight bounding box in case of pedestrian videos taken from TUD-Stadtmitte dataset [8].

Pose Annotation in Videos. Estimating 2D human pose from images is a challenging task with many applications in computer vision, such as motion capture, sign language, human-computer interaction and activity recognition. Figure 1.2 shows examples of 2D human pose in images. It shows a full body human pose estimation for 14 joints i.e., head, neck, left-right shoulder, left-right elbow, left-right wrist, left-right hip, left-right knee and left-right ankle. Human pose estimation is one of the classical problems in the computer vision but often the predictions are absurdly erroneous in videos due to unusual poses, challenging illumination, blur, self-occlusions etc. These erroneous predictions can

be refined by leveraging previous and future predictions as the temporal smoothness constrain in the videos. We present a generic approach for pose correction in videos using sequence learning that makes minimal assumptions on the sequence structure.

Figure 1.2: Example showing full body human pose or skeleton with 14 key-points annotation i.e., head, neck, left-right shoulder, left-right elbow, left-right wrist, left-right hip, left-right knee and left-right ankle in 2D images

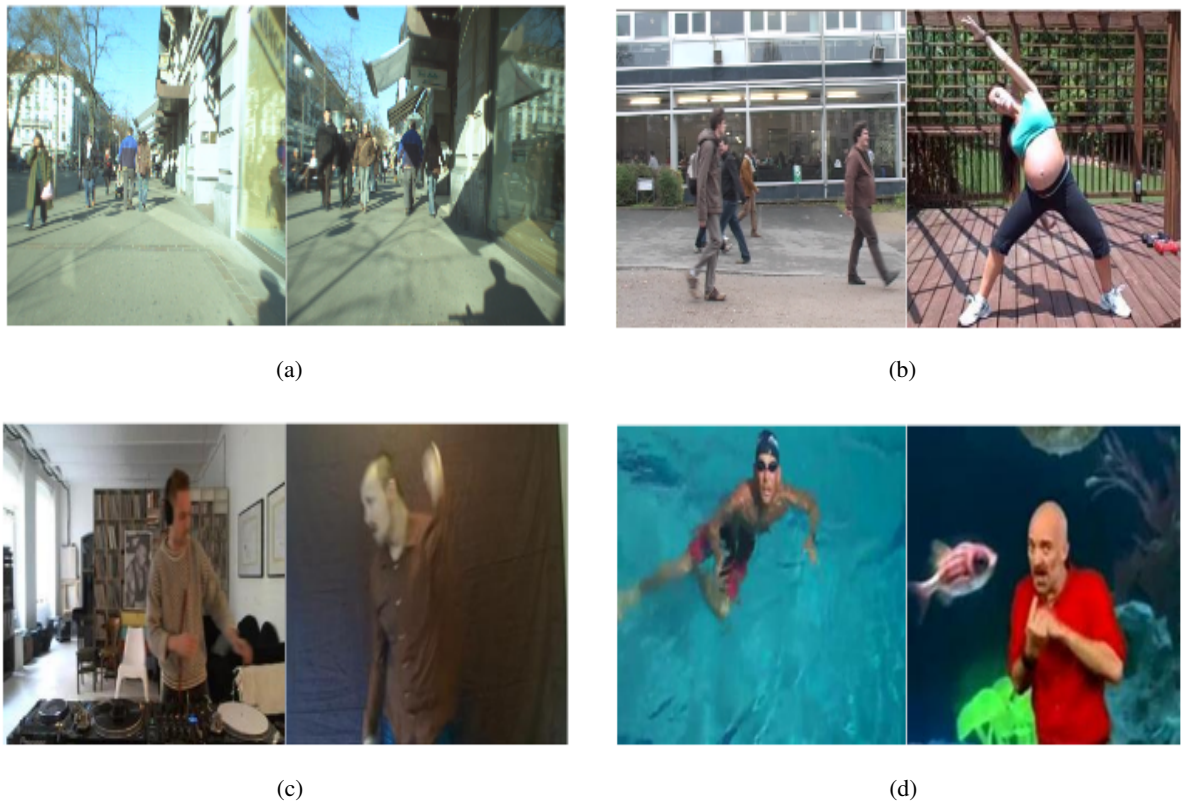
1.2 Motivation

More than 1 Billion unique users visit YouTube each month, watching 6 billion hours of video, and uploading 100 hours of video every minute. Cameras are now ubiquitous and sifting through this ocean of data has become a major global challenge.

The widespread proliferation of digital media, and in particular video data, in recent years, has made the automated analysis of its content necessary for annotation, retrieval, storage, transmission, security and commercial purposes. Video is analyzed for the tracking, detection and recognition of human activities in a variety of setups, ranging from constrained indoors environments to outdoors locations and videos in the wild. Applications include health monitoring, analysis of online content, annotations, effective classification. It is also analyzed for the detection of unknown, unusual events and abnormalities using and developing computer vision, machine learning and statistical methods. Applications include security and surveillance applications, in the case of traffic or crowd videos, but also more general videos where abnormal events may occur.

1.2.1 Applications

- **Object Tracking.** Object tracking can be described as a correspondence problem, and involves finding the relation between objects in contiguous frames.



(a) (b)

(c) (d)

Figure 1.3: Challenges. (a) Illumination Changes, (b) Occlusion (External and Self), (c) Motion Blur and (d) Dynamic Background. The examples are taken from ETH-SunnyDay [32], ETH-JBanhof [32] and YouTube Pose dataset [23].

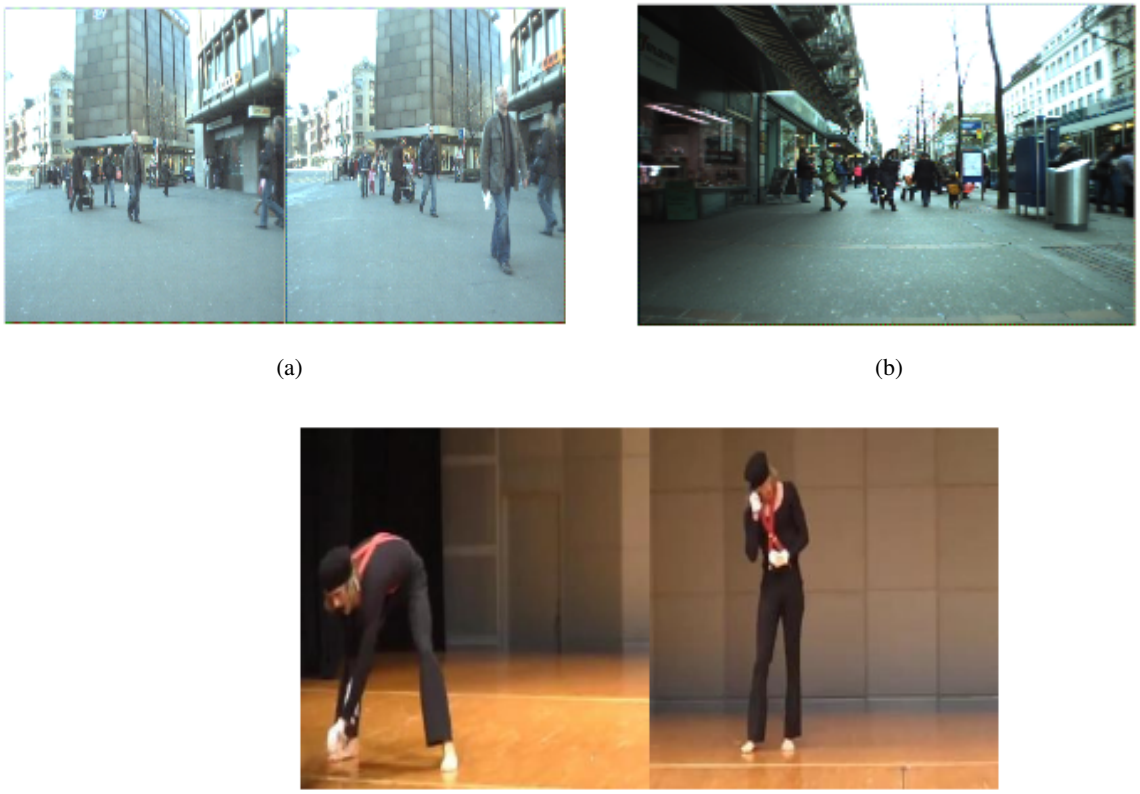
- **Event detection.** Event detection can be used to identify the events in video and has a variety of applications like action classification, security systems (raise alarms if events are suspicious), video understanding, summarization.

- **Motion Detection.** Motion detection algorithms are the basis for a wide range of applications in computer vision like visual surveillance, object recognition and tracking and compression of video streams.

1.3 Challenges

Video data poses a variety of challenges which include:

- **Illumination changes.** Lighting conditions in outdoor images vary drastically. They may range from very high to very low foreground-background contrast. This might result in highlighting random objects in the scene. Indoor surroundings and illumination can be manipulated to make



(a) (b)

(c)

Figure 1.4: Challenges. (a) Camera Movement and object speed, (b) Background Clutter and (c) Shape Variation. The examples are taken from ETH-Jelmoli [32], ETH-JBanhof [32] and YouTube Pose dataset [23].

the object of interest in focus with respect to the background. Illumination changes usually results in challenging problems for many computer vision applications such as recognition, tracking and motion analysis. Figure 1.3 (a) shows varying illumination conditions highlighting the background or the foreground obscuring visibility of complete human.

- **Occlusion (External & Self).** In general, outdoor images have more than one object or the object interacts with elements of the surroundings. This leads to occlusion by some other object like a pole, building etc. These cases like partial or full occlusion create uncertainty in determining the location of object which are not visible. Apart from the above mentioned occlusions which were caused by an outside entity (whether living or non-living object), the object of interest can occlude itself, which is termed as self occlusion. Figure 1.3 (b) shows such examples of self and external occlusion. Self-occlusion is observed mostly in outdoor scenarios such as sports, dance, exercises etc.

- **Motion Blur.** This is a widely occurred phenomenon in videos. For example, videos in which, human performs speedy actions like dancing, the limbs are subjected to move rapidly. Even with a high frames per second (fps) extraction, the dislocation of corresponding parts is of a high degree. Also the tracking strategy methods like Optical Flow or SIFT Flow will not be able to grasp the possibilities unless a high quality part tracker is used. Figure 1.3 (c) shows artifacts like motion blur in fast moving parts.

- **Dynamic Background.** Some parts of the scene may contain movement (a fountain, movement of other objects, the swaying of tree branches, water waves etc.), but should be regarded as background, according to their relevance. Such movement can be periodical or irregular (e.g., traffic lights, waving trees). Handling such background dynamics is a challenging task. Figure 1.3 (d) shows examples of Dynamic background where water waves have irregular motion.

- **Camera Movement.** Videos captured by a moving camera or an unstable (e.g. vibrating) camera, induce jitter in videos. The nature of noise introduced by jitter is sporadic. In case of moving camera, the background is not static and the objects move in the same or opposite direction of camera motion, hence changing the object size and appearance accordingly. Figure 1.4 (a) shows examples where the motion of camera and the person are in opposite direction, hence the object size increases as the camera moves closer.

- **Background Clutter.** Backgrounds can be highly complex which can make foreground background segmentation a really tough task even for the human eye. Multi-patterned and colored background is probable to excite the filters being used at various locations leading to a large number of false positives. Figure 1.4 (b) shows an example of a scene where the background and foreground are almost inseparable.

- **Shape Variation.** Another kind of challenge faced because of complex human pose configuration is foreshortening of body parts especially limbs. Figure 1.4 (c) shows examples of variation in shape when a person is dancing. It can be observed that as the person bends, there is a huge difference in terms of object shape.

- **Appearance diversity.** The diversity in clothing different people wear is difficult to encapsulate in a single model. Most of the times the silhouette of a person is the detailing added by the clothing of the person. Some clothing types cover a number of body parts making it difficult to estimate. The images shown in Figure 1.3 and 1.4 show the variations in appearance.

- **Object Speed.** The speed of the moving object plays an important role in its detection. Intermittent motions of objects cause ghosting artifacts in the detected motion, i.e., objects move, then stop for a short while, after which they start moving again. There may be situations in a video,

where the still objects suddenly start moving, e.g., a parked vehicle driving away, and also abandoned objects. Figure 1.4 (c) shows a person dancing and the rate at which the person changes position varies according to the song.

1.4 Our Contributions

The specific contributions of the work discussed in this thesis are as follows:

- **Efficient Object Annotation for Surveillance and Automotive Applications.** We focus on efficient object annotations (tight bounding boxes) for surveillance and automotive applications. We focus on one of the aspects of video annotation namely the annotation propagation. This step plays a vital role in reducing the cognition load by incorporating human inputs - in the form of annotations - at certain frames and automatically transferring them to the neighboring frames. This eliminates the need to manually annotate every frame in the video. An energy minimization scheme for the segmentation is the central component of our method. As we use segmentation to generate the annotations, the resultant bounding boxes are tight with high recall.
- **Human Pose Correction using Sequence to Sequence learning in Videos.** Goal is to generate accurate pose annotations by leveraging previous and future predictions as the temporal smoothness constrain in videos. We propose a generic approach for pose refinement using sequence-to-sequence learning which makes minimal assumptions regarding the sequence structure. The proposed architecture uses Long Short-Term Memory (LSTM) encoder-decoder model to encode the temporal context and refine the estimations.

1.5 Thesis Outline

The organization of the thesis is as follows. In the following chapter, we discuss object annotations in detail. In chapter 3, we propose a semi-automatic initialization scheme to annotate object bounding box in large scale complex videos. We annotate tight bounding boxes efficiently by using motion cues and dynamic GMM update in videos. In chapter 4, an annotation scheme for human pose in videos is proposed. Using a completely automatic initialization with the help of computer vision algorithms and the temporal information in videos we propose an architecture to annotate the erroneous poses.

Chapter 2

Object Annotations

Data Annotation plays a crucial role as they serve as a reference (ground truth) for supervised learning in any domain for problem solving. In the current era of deep learning, large amount of data is required to train the networks, which requires a collection of large amount of data which itself is a difficult task. Getting the dataset annotated, with minimal errors is also a big challenge. The error in the annotation is dependent on the complexity of annotation which in turn depends on the category of annotation. For example, in case of dataset annotation for the task of classification, the annotation would be a class label from a pre-defined set. It indicates the classes present in the image. If the task is human pose estimation, then the data will be annotated with the key-points i.e., head, neck, shoulders etc. If we compare the possibility of annotation error, then highly likely it would be more in case of human pose estimation over classification, as it requires the key-point annotations to be precise at the pixel level.

The goal is to minimize the annotation efforts by automating the processes with the help of vision and machine learning algorithms, to strive for more accurate annotations with minimal cost. But since the annotation complexity varies from problem to problem, it is required to analyze these annotations independently. In next section, we discuss in detail the types of annotations and available datasets for each case.

Figure 2.1: Sample image taken from Imagenet [28] with object classification Annotation

There are a wide variety of object annotations available. The annotations were designed in order to solve the problem of interest. Few of the object annotations include:

Class Labels. Class label annotation is one of the primary annotation, which assigns a label for each class. It is used to solve classification related problems. In classification problems, the task is to identify objects in images or actions in videos respectively. For example, in video classification, the goal is to identify the action (one of the classes in the dataset) in a video. Table 2.1 lists a few datasets with class labels as annotations.

Dataset Name	Description	Format
ImageNet [28]	Labeled object image database	Images
MNIST [8]	images of digits	Images
CIFAR [6]	natural images	Images
Caltech 256 [44]	pictures of objects	Images
Pascal VOC [33]	images of objects	Images
MS COCO [63]	Images of natural scenes	Images
Sports-1M [54]	sports videos	Videos
YouTube-8M [5]	youtube videos	Videos

Table 2.1: List of datasets with Class Labels as annotations.

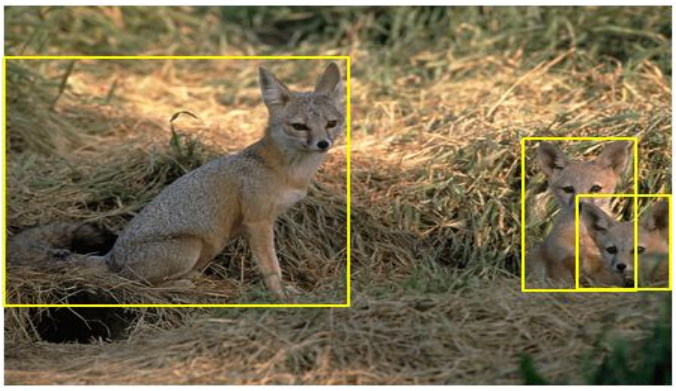


Figure 2.2: Sample image taken from Imagenet [28] with Bounding Box Annotation of object Cat

Bounding Box. The class labels give information about the objects in the scene but they do not describe about their position in the scene. To describe the object position in the image, one can draw a

Dataset Name	Description	Format
TUD-Stadtmitte [8]	static camera & pedestrians	Videos
TUD-Campus [6]	static camera & pedestrians	Videos
TUD-Crossing [6]	static camera & road crossing	Videos
ETH-Person [32]	moving camera & urban scene	Videos
Caltech 256 [44]	pictures of objects	Images
FDDB [48]	face images	Images
WIDER FACE [94]	face images	Images
GTFD [69]	face images	Images
ImageNet [28]	Labeled object image database	Images
Pascal VOC [33]	images of objects	Images
YouTube-BB [76]	youtube videos	Videos
TDCB [62]	street images with cyclists	Images
VGG Face [71]	face images	Images
STWD [45]	synthetic images	Images

Table 2.2: List of datasets with Bounding Box annotations.

rectangle around the object to know it's position. Hence, the object is annotated with a bounding box such that the object lies completely inside the box. A Bounding box is either represented by upper-left coordinates, width and height of box or upper-left and bottom-right coordinates. Bounding box annotations are used for object detection, object tracking, object recognition, scene understanding, image captioning, summary etc.

Table 2.2 shows a list of few datasets which have bounding box annotations. Figure 2.2 shows example of bounding box annotation in an image for the object "cat". It can be observed that the bounding boxes are loose i.e., there more pixels of background (not belonging to object) inside the bounding box.

Semantic Labeling. Although, the bounding box provides information regarding the position of object in the image, but it is not precise. Often it is a loose bounding box, for example if we use bounding

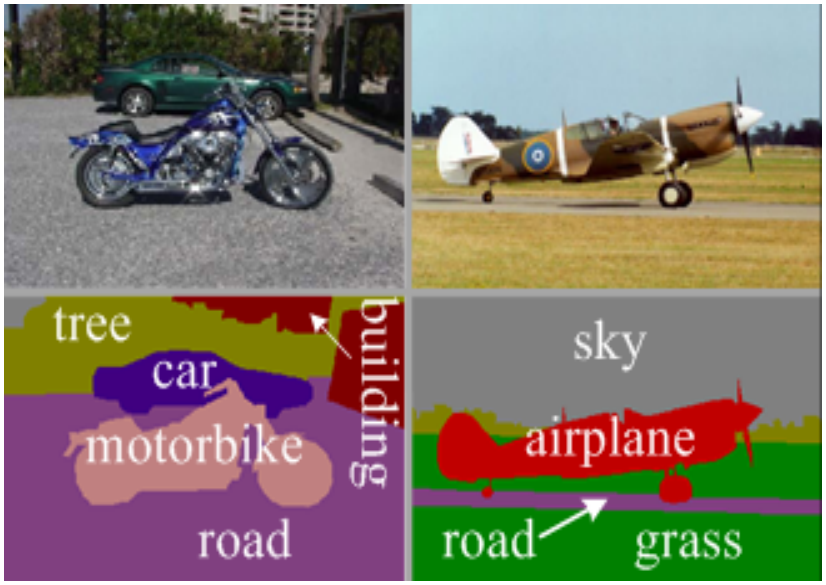


Figure 2.3: Example image taken from MSRC-21 [86] showing the semantic labeling Annotation of natural images.

box annotation to annotate a cat, almost half of the pixels in the box will belong to the background. For a more accurate description of object, objects can be described at the pixel level. Semantic labeling annotates objects at pixel level, it annotates each pixel as object (i.e., 1) or background (i.e., 0) in case of only foreground-background segmentation. In case of multiple objects, each pixel is assigned to a class. Table 2.3 shows a list of few datasets which have semantic labeling annotations. These type of annotations provide accurate object details and help in better scene understanding.

Figure 2.4: Sample images showing the full body human pose annotation taken from Leeds Sports & extended dataset [52] and MPII Human Pose dataset [9].

For example, the cityscapes [27] dataset has street images with pixel wise annotations, giving detailed information about every object like tree, car, road etc. Figure 2.3 shows examples of natural images

Dataset Name	Description	Format
CityScapes [27]	street scenes	Images & Videos
MSRC [86]	images of real objects	Images
Pascal VOC [33]	images of objects	Images
MS COCO [63]	Images of natural scenes	Images
MDRS3 [78]	road scenes	Images
Synthia [77]	synthetic images of urban scenes	Images
DAVIS [72]	natural scenes	Videos
BSD [65]	images of objects	Images
Flowers Dataset [70]	flower images	images
ParisSculpt360 [10]	paris images	Images
OpenSurface [15]	real world scenes	Images
CamVid [20]	urban scenes	Videos
Virtual KITTI [38]	photo-realistic synthetic videos	Videos
NYU-Depth V2 [68]	RGBD indoor scenes	Videos

Table 2.3: List of datasets with Semantic Labeling annotations.

and their corresponding semantic labeling annotations, as shown it labels every pixel as any one of the given objects.

Human Pose. Semantic labeling provides pixel-level detail about the scene but it fails to provide information regarding the object. For example, a human can be annotated at pixel-level using semantic labeling, but information regarding human(for example posture, limb locations) is not known. Knowing posture of human helps in semantically reasoning about the scene and will also help in solving other problems like cloth parsing, action recognition etc. Human posture is nothing but the human layout that is the skeleton, it can be represented by joint positions. For a full body human pose, a person is annotated with 14 key-points that constitute to form skeleton (e.g., head, neck, left-right shoulder, left-right elbow, left-right wrist, left-right hip, left-right knee and left-right ankle).

Dataset Name	Description	Format
LSP & extended [52]	Sports, athletics images	Images
FLIC [81]	Images from movies	Images
Buffy [37]	Images from TV show buffy	Videos
Buffy 2 [49]	images from TV show buffy	Images
MPII Human Pose [9]	natural images	Images
ETHZ Pascal [31]	amateur photographs	Images
Poses in the wild [25]	natural images	Images
CVIT Sports [87]	sports images	Videos
Football I & II [57]	sports images	Images
YouTube Pose [23]	amateur youtube videos	Videos
BBC Pose & extended [74]	BBC videos with sign language signers	Videos
JHMDB [51]	action clips	Videos
Movie Stickmen [49]	images from hollywood movies	Images
H3D [17]	natural images	Images

Table 2.4: List of datasets with Human Pose annotations.

Table 2.4 shows a list of few datasets which have human pose key-point annotations. Human pose can help in reasoning and solve many other problems. For example, in cloth parsing accessories like belt will be around hip and tie will start from neck till waist (which are known from the human pose).

Annotating human pose will give information about the person i.e., the position of head, wrist and so on etc. But concise details regarding some body parts like the hand (gesture or pose), face (viewpoint) is not known. For example, using human pose, we get to the position of wrists, but the hand gesture (fingers and their positions) is not captured.

Hand Pose. To capture the hand pose, 21 hand-joint positions are annotated. Table 2.5 shows few datasets which capture hand pose annotations. By capturing hand pose, we can know the position of each and every finger joint, which has many applications. A few applications would be sign-language recognition, augment reality games, hand gesture recognition etc.

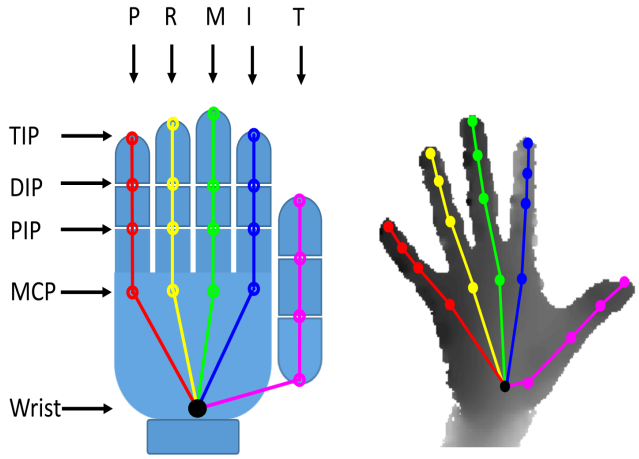


Figure 2.5: Image showing the 21 key-points of hand pose annotation where 'T', 'I, 'M, 'R, 'P denote 'Thumb, 'Index, 'Middle, 'Ring, 'Pinky fingers taken from [96].

Figure 2.5 shows the 21 hand-joint positions which are captured for annotation. The joints are ordered in this way: Wrist, TMCP, IMCP, MMCP, RMCP, PMCP, TPIP, TDIP, TTIP, IPIP, IDIP, ITIP, MPIP, MDIP, MTIP, RPIP, RDIP, RTIP, PPIP, PDIP, PTIP, where 'T', 'I, 'M, 'R, 'P denote 'Thumb, 'Index, 'Middle, 'Ring, 'Pinky fingers. 'MCP, 'PIP, 'DIP, 'TIP as in the figure 2.5.

Dataset Name	Description	Format
NYU-Hand pose [90]	RGBD hand images	Images
Big Hand 2.2M [96]	hand images using 6D sensors	Images
ICVL [89]	RGBD hand images	Images
MSRC [84]	RGBD hand Synthetic images	Images
SHPTB [99]	RGBD hand videos	Videos

Table 2.5: List of datasets with Hand Pose annotations.

Head Pose. To capture the head pose of person, yaw, pitch and roll are required to be annotated. Figure 2.6 show the yaw, pitch and roll along Y, X and Z axis respectively. Capturing head pose is important in some applications and is a part of other problems like capturing eye gaze. For example, to score the driver's driving attention in videos, the driver attention score depends on the head pose. Table 2.6 shows few datasets which caputre the head pose annotations.

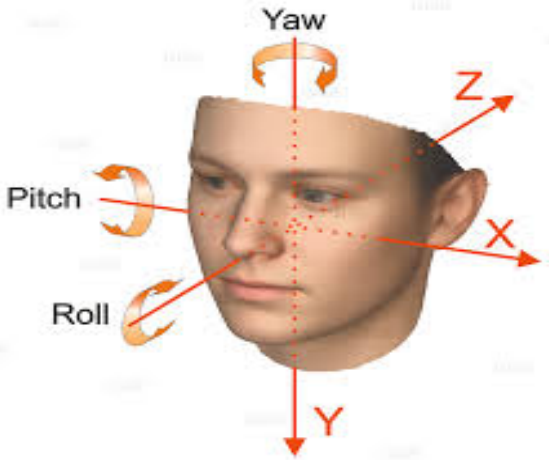


Figure 2.6: Sample of Head Pose Annotation

Dataset Name	Description	Format
UPNA Head Pose [11]	head tracking and pose videos	Videos
HPID [40]	images of faces	Images
HPED [98]	images of faces	Images
AFW [100]	face images	Images
AFLW [67]	face images	Images

Table 2.6: List of datasets with Head Pose annotations.

Facial Points. Head pose captures the head position but it does not provide detailed information of face like eyes, nose, lips, cheeks etc. To capture the facial expressions, it is required to annotate the facial points. Table 2.7 shows a list of few datasets which have facial points annotations. Figure 2.7 shows example of 68 facial landmarks annotated on a sample face image. These facial landmarks can be used for identification, gesture recognition, person mood estimation etc.

Other Annotations. Apart from the above discussed annotations, other annotations include 3D object representations (like 3D shape, 3D human pose etc), eye gaze, lip reading (associates speaker utterances to words), scene summary etc. For example given an image, the scene summary annotation would be a sentence describing the image (example: A cat is under the table). The applications include visual-dialog system, image captioning. Extending it to videos, the annotation would be a video summary (retaining only the important segments in video).

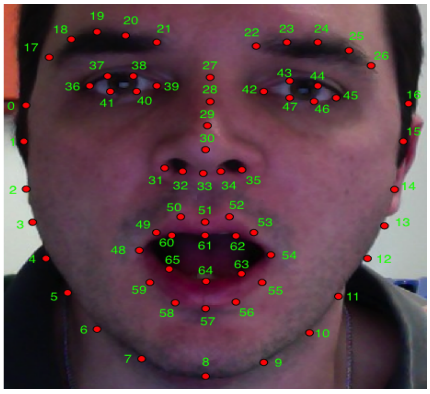


Figure 2.7: Sample of Facial Landmark Annotation

Dataset Name	Description	Format
Helen & extended [60]	face images	Images
LFPW [14]	face images	Images
LFW [46]	face images	Images
SCFace [42]	face images	Images
COFW [21]	face images	Images
ibug 300W [80]	face images	Images
ibug 300VW [85]	face images	Videos
AFLW [67]	face images	Images
CAS-PEAL [39]	face images	Images
PUT Face [55]	face images	Images

Table 2.7: List of datasets with Facial Point annotations.

Chapter 3

Efficient Object Annotation in Videos

3.1 Introduction

Accurately annotated large video data is critical for the development of reliable surveillance and automotive related vision solutions. In this work, we propose an efficient and yet accurate annotation scheme for objects in videos (pedestrians in this case) with minimal supervision. We annotate objects with tight bounding boxes. We propagate the annotations across the frames with a self training based approach. An energy minimization scheme for the segmentation is the central component of our method. Unlike the popular grab cut like segmentation schemes, we demand minimal user intervention. Since our annotation is built on an accurate segmentation, our bounding boxes are tight. We validate the performance of our approach on multiple publicly available datasets.

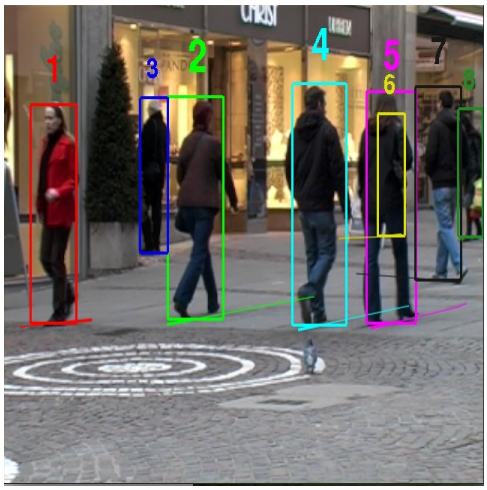


Figure 3.1: Sample Bounding Box Annotation of objects

In the recent years researchers have developed and demonstrated robust computer vision methods based on supervised machine learning techniques. They show impressive results on tasks such as large scale object detection, object recognition and image classification [4, 34, 58, 79, 83], reinstating the

hope that they are nearing mainstream adaptation. Most of these methods find applications in fields such as image search, web based face recognition engines [88] and other applications [61]. However these methods are far from mature when it comes to deploying in mission critical applications such as in surveillance, robotics, and autonomous driving. Accuracy and reliability of the vision algorithms need further improvement.

Since the state-of-the-art computer vision schemes depend on supervised learning techniques, such a deficiency in performance can be attributed to the following reasons. Firstly, it is difficult to annotate and generate large sets of training data covering a wide gamut of foreseeable working conditions which is necessary for supervised learning algorithms to generalize. The second hurdle is to generate a large amount of validation and test datasets that can help to find the model complexity and benchmark the performance. For example, one needs to test a solution for several hundreds of hours to reliably estimate the performance in an autonomous driving to be practical. Finally, the complex models that can be trained with the computational resources and evaluated on a wide range of hardware. There have been many success stories of deep learning in recent years which is known to be data intensive. A common underlying component is the generation of large scale reference data, also known as Ground Truth (GT). We are interested in this. In this work we restrict the scope of GT to a bounding box enclosing the spatial extent of articulate objects (humans) within a video frame. They are the most common and vulnerable subjects in the context of surveillance and autonomous driving scenarios. There have been many attempts in generating annotated data in computer vision in the past (eg. ImageNet). However, large industrial scale annotation efforts are not often reported in the literature.

In most cases GT is generated via manual annotators [35, 43, 66] who mark the object of interest in a set of images. More recently researchers have started to use video sequences to generate GT as they provide much richer representations of the object [1, 56]. As pointed out in [93], manual annotation especially in videos involves a huge cognition load, and is subject to inefficiency and inaccuracies. This is more evident while annotating humans as the limbs might move in a nonlinear manner and is difficult to capture the resulting variation in shape and extent of object within neighboring frames. Recently researchers have engineered various approaches to address the above aspects. Some of them are based on crowd-sourcing [1] and others use computer vision and machine learning techniques to develop semi-automatic annotation methods [2, 3, 16, 29, 56, 97]

In this work we focus on one of the aspects of video annotation namely the annotation propagation. This step plays a vital role in reducing the cognition load by incorporating human inputs - in the form of annotations - at certain frames and automatically transferring them to the neighboring frames. This eliminates the need to manually annotate every frame in the video. Our annotation propagation method involves segmenting an object and propagating the segmentation mask across the frames.

3.2 Video Annotation Tools

There have been many attempts in annotating images and videos in the past. Researchers from IBM developed a video annotation tool namely VideoAnnEX [3]. The objective of the tool is to generate GT, so as to facilitate the process of video/information retrieval based on certain user query (e.g., provide me all the frames/information related to a particular public figure). This uses a supervised learning method to label scenes in the video but lacks object level annotation and propagation.

ViPER [2, 29] is an open-source tool developed by Language and Media Processing lab (LAMP) at University of Maryland to generate GT in video sequences and also an evaluation framework to evaluate performance of algorithms for tracking, recognition and detection. To speed up annotation process ViPER provides a mechanism to interpolate bounding box position of objects in between key frames. In addition to this it is also possible to propagate object specific attributes between key frames by using the copy functionality or dragging across video frames. Although ViPER has annotation propagation method, in general it is not effective and scalable while generating GT of articulated objects as the accuracy of propagation is directly proportional to the granularity of key frame interval and nature of object motion which in our case is nonlinear (e.g. limbs and legs). In other words, it would be difficult to fully capture the limbs of a person moving across the camera 's field of view.

LabelMe video (LMV) [97] is another open source web accessible video annotation system that allows to annotate object category, shape, motion, and interactions between them. It is an extension of the popular LabelMe image annotation tool. Here the GT is generated by automatically propagating manual annotation across neighboring frames with the help of offline recorded camera motion parameters. These parameters are used to estimate homography between adjacent frames which is then used to accurately propagate the annotation. Although the method works reasonably well, in most cases it might not be practical as it would involve expensive additional hardware to record precise camera motion parameters.

The authors of Innovative Web based collaborative platform for video annotation [56] provide ways to share annotations through a collaborative web based platform. In addition the tool derives annotations based on annotations from multiple users. Also the tool provides GrabCut based features to extract boundaries of an object. The major limiting factor of this tool is that it doesn't involve propagation of object boundaries or GT across frames.

Another web based video annotation tool known as VATIC [1] is based on crowd-sourcing and developed at University of California at Irvine with collaborations from Massachusetts Institute of Technology. The tool is in experimental stages and was hosted on amazon's mechanical turk to investigate the potential of crowd sourcing platforms for the task of video annotation. The tool incorporates active learning based methods to propagate annotation across frames. However, the authors do not place much importance on the accuracy of an object's spatial extent and hence might not readily fit our purpose to generate GT for mission critical applications and articulated objects.

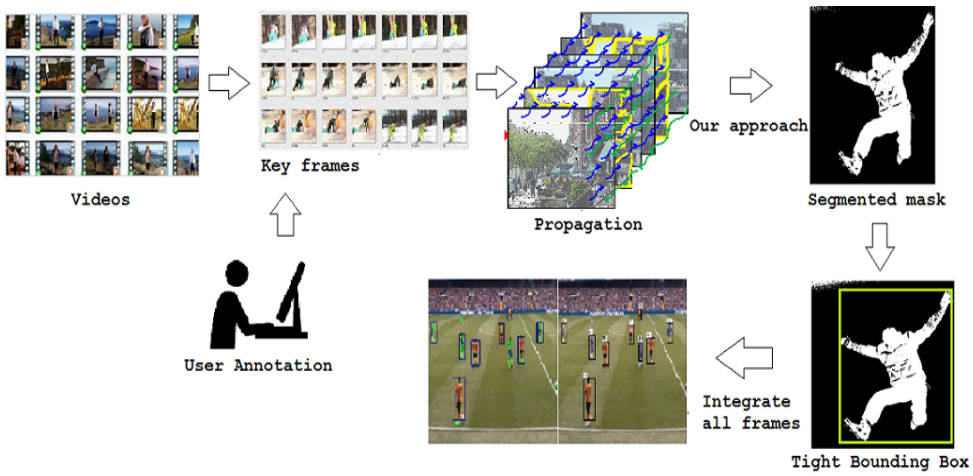


Figure 3.2: Proposed framework: Given a set of videos, user annotates selected key frames. We propagate the annotations for the entire sequence and use it as initialization for our approach and get a tight bounding box.

An interesting video annotation tool named as iVAT [16] incorporates incremental learning based approach to build online detector that aids linear interpolation and template matching based annotation propagation across video frames. The role of the detector is mainly to resolve occlusions and handle any non-uniform/non-linear motion of an object. However, the tool doesn't determine the spatial extent of an object in each frame and hence may significantly affect the accuracy of GT in our case.

As a complementary component to the existing tools, we propose a method where annotations on key frames are propagated automatically using motion cues. We adopt GrabCut [22] based segmentation method for videos to obtain highly accurate annotations for objects in large scale videos efficiently. GrabCut is a successful interactive segmentation scheme. We adapt it for our problem so that it demands very minimal user intervention (say on selected key frames). The advantages of our method are two fold, firstly we only need user interactions on key frames, and hence it reduces the human effort drastically. Secondly, since our method tracks the object using segmentation, we obtain more accurate bounding boxes around objects.

3.3 Proposed Approach

Given a large collection of videos our goal is to design a method which can automatically propagate the annotations for the objects in those videos with very minimal user interaction. We propose a method for accurate propagation of annotations of objects in videos using segmentation. Our proposed scheme is illustrated in Figure 4.3.

For this our method is inspired by the success of GrabCut [22] and thereafter works such as obj-Cut [59] for natural image segmentation. Nevertheless, we propose several useful modification to the original GrabCut [22] to suit our problem. GrabCut has shown state-of-the-art image segmentation results on some of the standard benchmarks. But it has some limitations in real time applications for

videos. For example: (i) initialization is critical in GrabCut and often performed by user. On large collection of videos asking for user interaction in all frames (or most frames) is not a feasible solution. (ii) computing min cut on every frame is a costly operation. We resolve the above limitations with a semi-automatic approach and also reducing the computations.

In this section we first formulate the problem as energy minimization problem, briefly describe GrabCut method and discuss about our proposed semi-automatic initialization scheme.

3.3.1 Energy Minimization Framework

We formulate the problem of segmenting objects in video frames in an energy minimization framework. Segmentation of an image can be expressed as a vector of binary random variables

$\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each random variable X_i takes a label $x_i \in \{0, 1\}$ based on whether it is object or background.

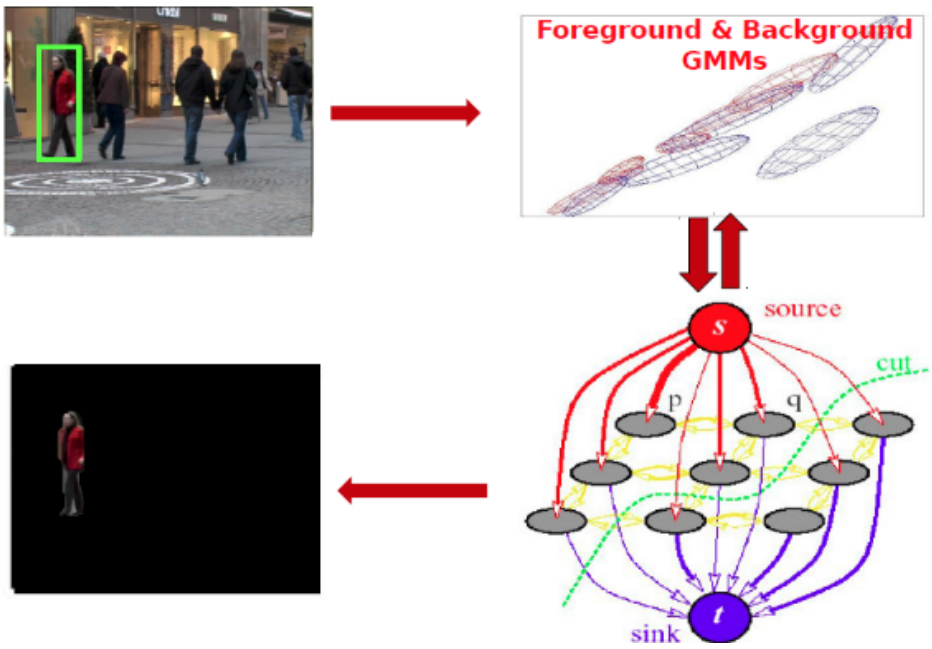


Figure 3.3: Energy Minimization Framework. Iteratively estimate the foreground and background GMMs and perform graphcut (graph min-cut). Given an image and bounding box which represents the object of interest, output is the object segmented from background.

We represent pixels of frames as nodes in a graph where all the neighbouring nodes are connected by edges. We associate a unary and pairwise cost of labeling these nodes and define a cost (or energy) function as the sum of these cost for all the nodes as follows:

$$\psi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i \psi_i(x_i, \boldsymbol{\theta}, z_i) + \sum_{(i,j)\in\mathbf{N}} \psi_{ij}(x_i, x_j, z_i, z_j), \tag{3.1}$$

where, **N** denotes the neighborhood system defined in the MRF, and ψ_i and ψ_{ij} correspond to unary and pairwise costs respectively.

A typical unary cost can be expressed as:

$$\psi_i(x_i, \boldsymbol{\theta}, z_i) = -\log p(x_i|z_i), \tag{3.2}$$

where z_i is the rgb color vector, $p(x_i|z_i)$ is the likelihood of pixel i taking label x_i. This likelihood is computed from learnt foreground and background GMM model. Reader is encouraged to refer [22] for more details. The pairwise cost is the given by [18]:

$$\psi_{ij}(x_i, x_j, z_i, z_j) = \lambda \frac{[x_i \neq x_j]}{ed(i,j)} \quad \exp\left(\beta(z_i - z_j)^2\right), \tag{3.3}$$

where the parameter λ controls the degree of smoothness, $ed(i,j)$ is the Euclidean distance between neighboring pixels i and j. The constant β allows edge-preserving smoothing, and is computed as follows: $\beta = 1/2\mathbb{E}[(z_i - z_j)^2]$, where $\mathbb{E}[u]$ is expected value of u.

The problem of segmentation is now to find the global minima of the cost function in 3.1, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \psi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}). \tag{3.4}$$

The global minima of this cost function can be efficiently computed by graph cut [19]. We use iterative graph cut based approach for computing this.

3.3.2 Semi-automatic Initialization

Given a set of video frames $\{f_1, f_2, \cdots, f_m\}$ we need to run iterative graph cuts in all the m frames. This is computationally not smart way. Moreover, foreground and background region needs to be initialized in each frame. In original GrabCut foreground and background are initialized by performing user interaction. However, we wish to avoid such user interactions to make our annotation process efficient and minimal human intensive. In other words we need two modifications from the original GrabCut method: (i) we wish to avoid performing iterative graph cuts in every frame, and (ii) we wish to minimize user interaction to great extent. To achieve this we propose following three automatic initialization schemes.

3.3.2.0.1 M1: Bi-linear interpolation. Given the annotation of key frames we extend the annotations to all the in-between frames by simply interpolating these annotations. These interpolated annotations are used to initialize the Gaussian mixture models of foreground and background in these frames.

3.3.2.0.2 M2: Relaxed interpolation. In the relaxed interpolation we extend the width and height of the interpolated bounding boxes by a small offset (w, h). Relaxed bounding boxes contain most or all the object inside them and achieve higher pixel level recall. Figure 3.4 shows an example of relaxed interpolation which is used as initialization for our method.

Figure 3.4: Semi-automatic Initialization. Extending the width and height of the interpolated bounding boxes by a small offset such that object lies inside the bounding box. **Orange** and **Green** bounding box represent interpolation prediction and extended bounding box.

3.3.2.0.3 M3: Using motion cues and dynamic GMM update. Each pixel is initialized with mixture of Gaussian mixture models (GMM). Using optical flow, we estimate the new position of each pixel. The new position can have different pixel value. Hence, we update the GMM for each frame dynamically.

Further, we estimate the number of Gaussian's to fit the model as the appearance changes across frames. This adapts to the incremental changes in the appearance of the object e.g., changes in illumination, object appearance. Assumption is moving pixels are considered as pixels belonging to the object or foreground. Again, there might be multiple moving objects, to get only the object of interest, we use the relaxed bounding box (M2) to eliminate the other pixels. Once we have an approximate estimate of the foreground pixels, the above estimated foreground mask is used as initialization seeds for GrabCut.

In brief, we propose the following simple but effective modifications for efficient segmentation and propagation to accurately annotate objects from the videos:

- We use relaxed interpolation to get the relaxed bounding box for the object of interest. It is calculated initially for the entire sequence (using the user annotations on key frames).
- Compute foreground and background GMM model at the key frames using the user annotations and these models are propagated to in-between frames.
- The above pre-computed models are used to segment object in nearby frames. As the object will be very similar in the nearby frames.
- The object neighborhood is sufficient to segment the object rather than taking the entire image. This reduces the computations drastically without effecting the results.

Figure 3.5: Sample results of segmentation by using motion cues and dynamic GMM update. (a), (e), (i) Image to be segmented and the green bounding box is used as initialization for segmentation, (b), (f), (j) shows the result of GrabCut on every frame , (c), (g), (k) shows the result of key frame segmentation and propagation and (d), (h), (l) shows the result of foreground estimation and segmentation.

Summarizing our approach, our method initiates with a very minimal user interaction (say drawing loose bounding box around objects on key-frames). We obtain annotations in the key-frames by segmenting them using iterative graph cuts. We propagate these annotations to the in-between frames using motion cues and dynamically update the GMMs for those frames. With these propagated annotations, segmentation and bounding boxes around all the objects in all the frames are obtained (refer figure 3.5).

(a) (b) (c)

(d) (e) (f)

Figure 3.6: Sample dataset images. Challenges: (a), (b) - change in object size, moving camera and moving object, (c), (d), (e), (f) - complex surroundings, occlusion and object pose variation.

3.4 Dataset

We have performed experiments on a variety of publicly available datasets for object tracking shown in Table 3.1. Each dataset has its own challenges, for example movement through cluttered areas, objects overlapping in the visual field, lighting changes, moving background, slow-moving objects, and objects being introduced or removed from the scene (refer Figure 3.6). For analysis, we selected a set of objects from these datasets with the above challenges. Sample images from the datasets we use are shown in Figure 3.6.

Figure 3.6 (a) and (b) show the case where a camera is moving and objects size change with time. Figure 3.6 (c), (d), (e) and (f) show the case where a camera is mounted and the objects are in motion (typically surveillance videos). All the cases have overlapping objects, dynamic entry and exit of objects, background clutter and objects moving with different speeds.

Dataset Name	Number of Objects	Number of Frames
TUD-Stadtmitte [8]	10	178
TUD-Campus [6]	9	71
TUD-Crossing [6]	13	201
ETH-Bahnhof [32]	224	999
ETH-Jelmoli [32]	74	936
ETH-SunnyDay [32]	36	354

Table 3.1: List of datasets. Each dataset has multiple annotated objects and the number of frames in each dataset is shown above.

3.5 Evaluation Protocol

There exists abundant performance measures in the field of object annotation in videos. We chose to use average area overlap and recall, as these performance measures the accurateness of annotations (as we aim for tight bounding box). We briefly introduce these measures here.

3.5.0.0.1 Average area overlap: It is the intersection area divided by union of ground truth (GT) and the bounding box (BB) generated by an annotation approach. The mean of this measure is calculated by dividing with total number of frames in the database.

$$AreaOverlap = \frac{1}{N}\sum_{k=1}^{n}\frac{Area(B_k^1 \cap B_k^2)}{Area(B_k^1 \cup B_k^2)} \tag{3.5}$$

where B_k^1, B_k^2 are bounding boxes of GT and annotation approach for k^{th} frame and N is the total number of frames.

3.5.0.0.2 Recall: Recall is computed as a fraction of true positive and true positive plus false negative, which are defined as follows.

$$TP = \sum_{k=1}^{n}\frac{Area(B_k^1 \cap B_k^2)}{Area(B_k^1 \cup B_k^2)} \tag{3.6}$$

$$FN = \sum_{k=1}^{n} \frac{Area(B_k^1)}{Area(B_k^1 \cup B_k^2)} - TP \tag{3.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

where B_k^1, B_k^2 are bounding boxes of GT and annotation approach for k^{th} frame, N is the total number of frames, TP, FN are True Positive and False Negative respectively.

3.6 Experiments and Results

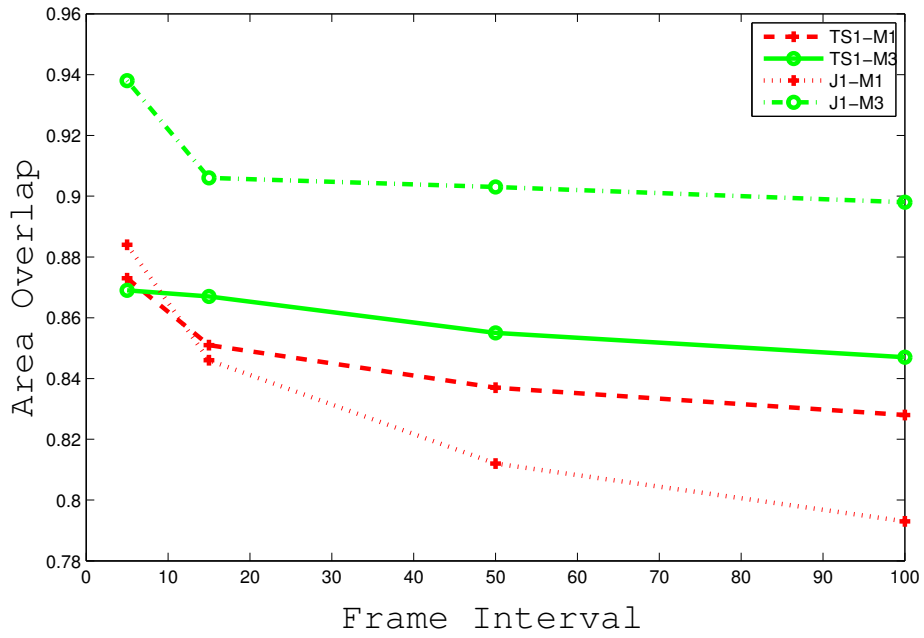
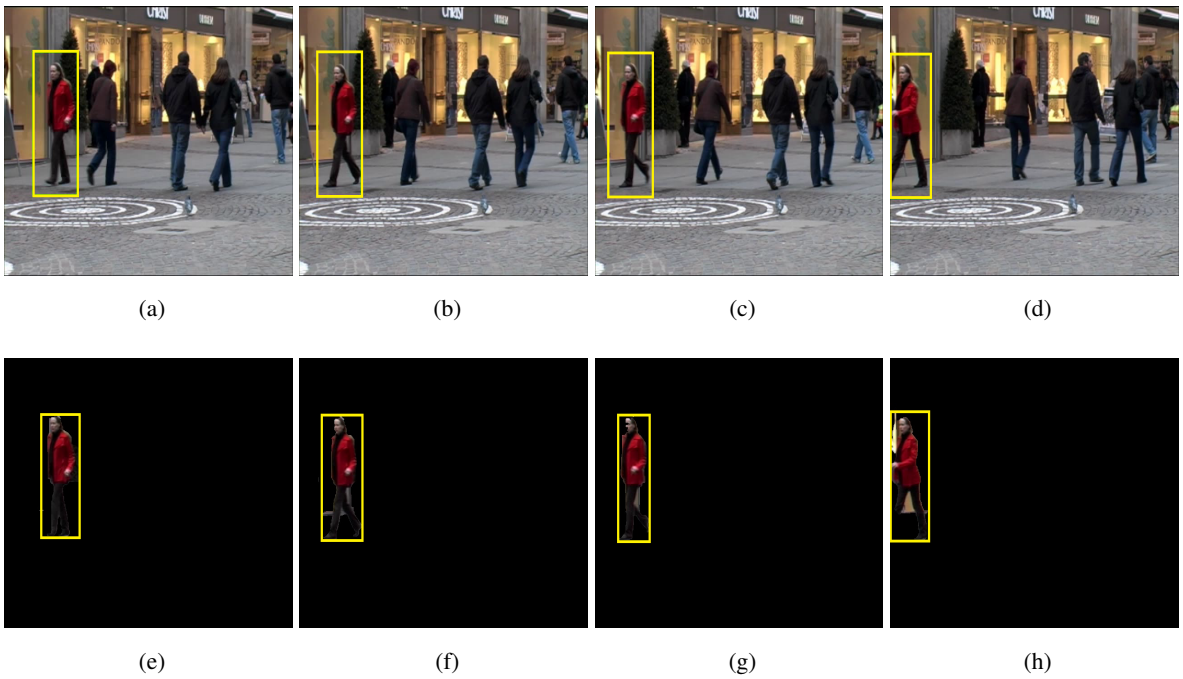


Figure 3.7: Frame interval vs Area overlap for the objects TS1: TUD-Stadtmitte 1 and J1:ETH-Jelmoli1. Red and green lines correspond to M1 and M3 respectively. It can be seen that the area overlap of our method (M3) increases indicating hih recall over M1.

We have applied the proposed approach on the datasets by varying the number of key frames, which accounts for user interactions. We have experimented the same in multiple settings by varying the key frame interval. The segmentation based method is not very successful in frames, where the user is occluded or improperly initialized (user initializes object where it is not completely visible). Also, we detect the frames where segmentation is unsuccessful and replace them with the interpolation result,



(a) (b) (c) (d)

(e) (f) (g) (h)

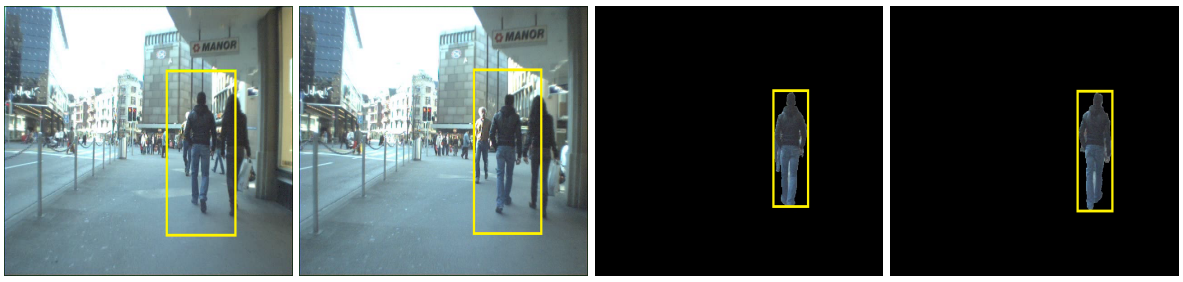
Figure 3.8: Results of our approach on object TUD-Stadtmitte 1. (a), (b), (c), (d) shows the initialization sequence and (e), (f), (g), (h) shows the result of our method respectively. We observe that the object is segmented accurately using our method which results in an accurate annotation.

which is not effected by occlusions. Hence, the proposed work clearly outperforms simple interpolation based technique.

Table 4.1 shows the area overlap and recall for objects with varying key frame intervals. Each object has a different context, TUD-Stadtmitte 1 object is for static camera, ETH-Jelmoli 1 object is for the case where both object and camera motion are in same direction and ETH-Banhof 3 object moves in opposite direction to the camera motion. From this table, we observe that (i) our proposed scheme M3 which uses motion cues performs better than M1 and M2, (ii) as expected with lower key frame interval we achieve higher recall and higher overlap (i.e., more accurate annotations).

Further, it should be noted that for ETH-Banhof 3, there are drastic changes in object size, due to movement in opposite direction. Hence, the errors are higher compared to TUD-Stadtmitte 1 and ETH-Jelmoli 1. Figure 3.7 shows the variation in area overlap for the methods discussed in Section 3.3.2 namely M1 and M3 with the change in frame interval for two different objects. To compare area overlap we use M1 and M3, as the annotations generated by M2 are relaxed bounding boxes, due to which the area overlap is less for M2. We also observe that for the object ETH-Jelmoli1, the area overlap for M3 is very high compared to M1 as our approach yields very accurate segmentations.

The proposed method achieves high area overlap and high recall using very few user annotations (refer Figure 3.8). We can see that our approach yields accurate segmentation which leads to accurate annotations. This is very helpful for large scale video annotations as we generate accurate annotations



(a) (b) (c) (d)

Figure 3.9: Results of our approach on object ETH-Jelmoli 1. (a), (b) show the initialization sequence and (c), (d) show the result of our method respectively.

with drastic reduction in the human annotation efforts. Consider an example where an user have a video of $10,000$ frames to be annotated. For a key frame interval of 10, user has to annotate only 1000 frames in our approach. This reduces the human efforts by 90 % without compromising on the accurateness of the annotations.

3.7 Summary

We have presented a framework for semi-automatic object annotation to generate accurate Ground Truth (GT) data in large scale from videos. Especially, our approach is suitable for generating GT for mission critical applications like surveillance and autonomous driving. Our method of object annotation is based on segmentation and its propagation which results in accurate bounding boxes around the objects. The proposed framework outperforms interpolation based approaches and almost mimics human annotation ability with only minimal user interaction (predominantly at key frames) which makes it scalable to generate large sets of GT . We have verified our claims by conducting comprehensive experiments on multiple challenging video datasets. Our approach can prove useful in generating ground truth and annotations for large scale surveillance and automotive related videos with substantial reduction in human efforts.

Object	Area Overlap			Recall		
	M1	M2	M3	M1	M2	M3
TUD-Stadmitte1 (5)	**0.873**	0.843	0.869	0.923	0.922	**0.931**
TUD-Stadmitte1 (15)	0.851	0.831	**0.867**	0.901	0.894	**0.927**
TUD-Stadmitte1 (50)	0.837	0.771	**0.855**	0.883	0.866	**0.906**
TUD-Stadmitte1 (100)	0.828	0.761	**0.847**	0.871	0.857	**0.896**
ETH-Jelmoli1 (5)	0.884	0.894	**0.938**	0.937	0.96	**0.965**
ETH-Jelmoli1 (15)	0.846	0.856	**0.906**	0.907	0.937	**0.963**
ETH-Jelmoli1 (50)	0.812	0.836	**0.903**	0.883	0.924	**0.963**
ETH-Jelmoli1 (100)	0.793	0.825	**0.898**	0.868	0.916	**0.964**
ETH-Banhof3 (5)	0.831	0.684	**0.84**	0.831	0.836	**0.847**
ETH-Banhof3 (15)	0.76	0.701	**0.77**	0.901	0.813	**0.907**
ETH-Banhof3 (50)	0.65	0.655	**0.67**	0.829	0.791	**0.834**
ETH-Banhof3 (100)	0.61	0.634	**0.65**	0.801	0.784	**0.903**

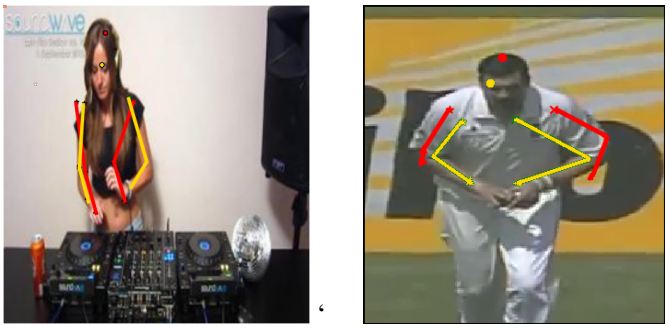
Table 3.2: M1: Bi-linear interpolation, M2: Relaxed interpolation, M3: Our approach which uses motion cues to propagate annotations (Section 3.3.2). The value in the parenthesis indicates the key frame interval.

Chapter 4

Pose Annotation in Videos

4.1 Introduction

The power of ConvNets has been demonstrated in a wide variety of vision tasks including pose estimation. But they often produce absurdly erroneous predictions in videos due to unusual poses, challenging illumination, blur, self-occlusions etc. These erroneous predictions can be refined by leveraging previous and future predictions as the temporal smoothness constrain in the videos. In this paper, we present a generic approach for pose correction in videos using sequence learning that makes minimal assumptions on the sequence structure. The proposed model is generic, fast and surpasses the state-of-the-art on benchmark datasets. We use a generic pose estimator for initial pose estimates, which are further refined using our method. The proposed architecture uses Long Short-Term Memory (LSTM) encoder-decoder model to encode the temporal context and refine the estimations. We show 3.7% gain over the baseline Yang & Ramanan (YR) [95] and 2.07% gain over Spatial Fusion Network (SFN) [73] on a new challenging YouTube Pose Subset dataset [23].



(a) (b)

Figure 4.1: Human Pose Correction. Pose predictions on sample images from the datasets used in this work. Red and yellow correspond to joints predicted using baseline (initialization) and our method respectively.

4.2 Pose Estimation Algorithms

Estimating 2D human pose from images is a challenging task with many applications in computer vision, such as motion capture, sign language, human-computer interaction and activity recognition. Profuse amount of work has been done on articulated pose estimation from single images [7, 30, 31, 36, 75, 82]. Despite steady advances, pose estimation remains as an intricate problem. Recent advances in 2D human pose estimation exploit complex appearance models and more recently convolutional neural networks (ConvNets) [24, 47, 50, 73, 74, 91, 92]. We focus on the task of 2D human pose estimation in videos "in the wild" : single-view, uncontrolled settings typical in movies, television and amateur videos. This task is made difficult by the considerable background clutter, camera movement, motion blur, poor contrast, body pose and shape variation, as well as illumination, clothing and appearance diversity. Even the state-of-the-art ConvNets often produce erroneous predictions in videos due to these challenges (Figure 4.1).

To date, CNN models for video processing have successfully considered learning of 3-D spatio-temporal filters over raw sequence data [12], and learning of frame-to-frame representations which incorporate instantaneous optic flow or trajectory-based models aggregated over fixed windows or video shot segments [54]. Such models explore two extrema of perceptual time-series representation learning: either learn a fully general time-varying weighting, or apply simple temporal pooling. Following the same inspiration, the video sequence learning models which are also deep over temporal dimensions; i.e., have temporal recurrence of latent variables. Recurrent Neural Network (RNN) models are "deep in time" - explicitly so when unrolled - and form implicit compositional representations in the time domain.

Pose predictions from neighbouring frames are not independent of each other and they form a sequence. We formulate the correction problem as sequence-to-sequence learning problem, while leveraging the temporal smoothness implicitly encoded in the target sequence. There have been a number of related attempts [13, 26, 41, 53] to address the general sequence-to-sequence learning problem with neural networks. Instead of correction of one prediction at a time (CRF based post processing), this model can capture the complex pose configurations over time while the body undergoes numerous appearance changes, resulting in a more reliable correction model.

In this work, we propose pose correction model in videos as a sequence-to-sequence learning problem (Figure 4.3). The neural network architecture, which we will refer to as an LSTM encoder-decoder, consists of two recurrent neural networks that act as an encoder and a decoder pair. The encoder maps an input source sequence to a fixed-length vector, and the decoder maps the vector representation back to a target sequence. The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence.

4.3 RNN Encoder-Decoder

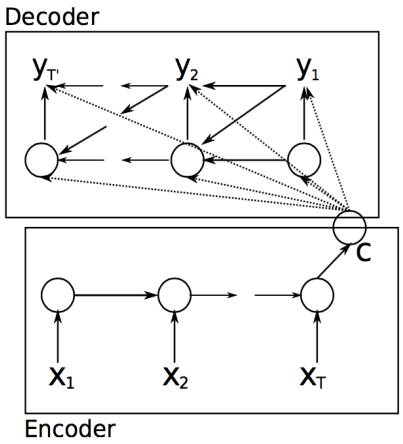


Figure 4.2: Illustration of RNN Encoder-Decoder model where $x_1, x_2, \cdots, x_t$ and $y_1, y_2, \cdots, y_t$ are input and output sequences respectively and c is the context vector.

In a recurrent neural network (RNN), for an input sequence $x = (x_1, \cdots, x_T)$ at each time step t, the hidden state of the RNN is updated by

$$h_t = f(x_t, h_{t-1}) \tag{4.1}$$

where f is a non-linear activation function. f can be a simple logistic sigmoid function or a complex Long Short-Term Memory (LSTM) cell.

By training RNN to predict the next symbol in a sequence, it can learn a probability distribution over a sequence. So, the output at each time step t is the conditional distribution $p(x_t|x_{t-1}, \cdots, x_1)$.

The encoder is an RNN which reads each token of the input sequence x sequentially. The hidden state of the RNN changes according to Eq. 4.1 as it reads tokens from input sequence. Once the entire sequence is read, the hidden state of the RNN is a context vector c of the whole input sequence.

Decoder is another RNN which is trained to generate the output sequence by predicting the next token y_t given the hidden state h_t. But, here both y_t and h_t are also conditioned on y_{t-1} and on the context vector c of the input sequence unlike the RNN described above. Hence, the hidden state of the decoder at time t is computed by,

$$h_t = f(h_{t-1}, y_{t-1}, c) \tag{4.2}$$

and the conditional distribution of next token is computed as,

$$P(y_t|y_{t-1}, \cdots, y_1, c) = g(h_t, y_{t-1}, c) \tag{4.3}$$

where f and g are activation functions.

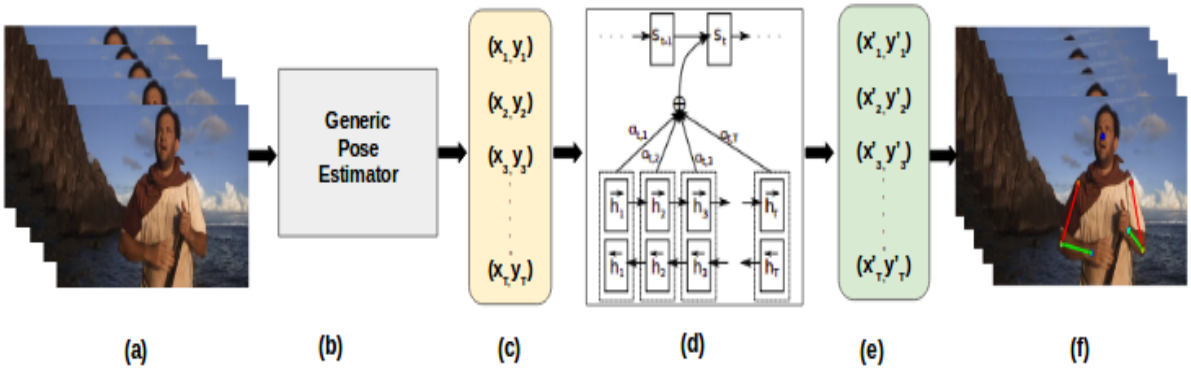


Figure 4.3: Overview of our method (a) Input videos, (b) Generic pose estimator, (c) Initial pose estimates (x_i, y_i) for all joints, (d) Correction model where h_t, s_t are the hidden states at time t and a_{ij} is an alignment model which scores how well the inputs around position j and the output at position i match, (e) Refined pose estimates (x'_i, y'_i) for all joints and (f) Pose visualization. A bidirectional LSTM encoder is used in the refinement model as shown in (d). The correction model corrects the erroneous poses (predicted by a generic pose estimator (b)).

Both components of RNN Encoder-Decoder are jointly trained to maximize the conditional log-likelihood

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y_n|x_n) \tag{4.4}$$

where θ is the set of the model parameters and each (x_n, y_n) is an (input, output) sequence pair from the training set.

4.4 Pose Correction Model

An overview of our algorithm is shown in Figure 4.3. Our approach can be broadly divided into 2 stages. Each stage is independent, and the details of each stage are discussed below.

4.4.1 Initialization

Our method receives frames from videos and generates initial pose estimates for all the frames independently. We can use any generic pose estimator to generate initial pose estimates. It is often observed that these estimates are erroneous in videos, due to self-occlusion, blur, unusual poses, etc (Figures 4.1 and 4.8). For our experiments, we use the Spatial Fusion Network (SFN) [73] and the more traditional Yang & Ramanan [95] models to generate initial pose estimates (discussed briefly in 4.7.1).

4.4.2 General Decoder

The conditional probability in Eq. 4.3 is defined as

$$p(y_i|y_{i-1}, \cdots, y_1, c) = g(y_{i-1}, h_i, c_i) \tag{4.5}$$

where h_i is an RNN hidden state for time i, computed by

$$h_i = f(h_{i-1}, y_{i-1}, c_i) \tag{4.6}$$

It should be noted that unlike the existing encoder-decoder approach (see Eq. 4.3), here the probability is conditioned on a distinct context vector c_i for each target y_i.

The context vector c_i depends on a sequence $(h_1, \cdots, h_{T_x})$ to which an encoder maps the input sentence. Each token h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i^{th} token of the input sequence. The context vector c_i is then computed as:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{4.7}$$

The weight α_{ij} of each h_j is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{4.8}$$

where

$$e_{ij} = a(h_{i-1}, h_j) \tag{4.9}$$

is an alignment model which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state h_{i-1} and h_j. The alignment model directly computes a soft alignment, which allows the gradient of the cost function to be backpropagated through. This gradient can be used to train the alignment model as well as the whole translation model jointly.

Let α_{ij} be the probability that the target token y_i is aligned to a source token x_j. Then, the i^{th} context vector c_i is the expected annotation over all the annotations with probabilities α_{ij}. The probability α_{ij}, or its associated energy e_{ij}, reflects the importance of h_j with respect to the previous hidden state h_{i-1} in deciding the next state h_i and generating y_i. Intuitively, this implements a mechanism of attention in the decoder. The decoder decides parts of the source sequence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sequence into a fixed length vector. With this approach the information can be spread throughout the sequence, which can be selectively retrieved by the decoder accordingly.

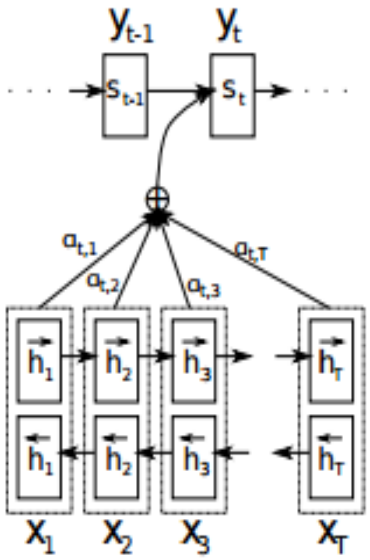


Figure 4.4: The graphical illustration of the model to generate the t-th target token y_t given a source sequence $(x_1, x_2, ..., x_T)$.

4.4.3 Bidirectional LSTM Encoder for pose correction

The usual RNN, described in Section 2, reads an input sequence x in order starting from the first symbol x_1 to the last one x_{T_x}. However, in the proposed scheme, we would like each word to summarize not only the preceding words, but also the following words. Hence, we use a bidirectional RNN, which has been successfully used recently in speech recognition (see, e.g., Graves *et al.* [41]).

For each token x_j, h_j is obtained by concatenating the forward hidden state $\vec{h}_j$ and the backward one $\overleftarrow{h}_j$. In this way, the h_j contains the summaries of both the preceding tokens and the following tokens. Due to the tendency of RNNs to better represent recent inputs, h_j will be focused on the words around x_j. This sequence is used by the decoder and the alignment model later to compute the context vector (Eqs. 4.7, 4.8).

We have initial pose estimates for each frame from the initialization stage. The correction model is trained to refine these sequences. We train the model to map the input pose sequence to target sequence (ground truth pose). There is a soft alignment between the input and output sequence elements. We now demonstrate the results of our approach on standard baselines and benchmark datasets.

4.5 Datasets

YouTube Pose. This new dataset consists of 50 videos of different people from YouTube by [23], each with a single person in the video. Videos range from approximately $2,000$ to $20,000$ frames in length. For each video, 100 frames were randomly selected and manually annotated ($5,000$ frames in total). Frames are annotated with 7 key-points i.e., head, left-right shoulder, left-right elbow, left-right

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.5: Sample images from the dataset. (a), (b), (c), (d) are sample images from YouTube Pose Subset dataset and (e), (f), (g), (h) are sample images from CVIT-Sports videos dataset.

wrist. The dataset covers a broad range of activities, e.g., dancing, stand-up comedy, how-to, sports, disk jockeys, performing arts and dancing sign language signers.

YouTube Pose Subset. A five video subset from YouTube Pose. The videos distribution for subset dataset is as follows: two disc jockeys, a mime artist, a dancing sign language signer, and one aerobics instructor.

CVIT-Sports. For our experiments, we use the CVIT-SPORTS- videos dataset by [87]. It is an extremely challenging dataset of humans playing sports. This set has a total of 11 videos of a human playing sports retrieved from YouTube. It includes intricate activities like cricket-bowling, cricket-batting, football. In total, this set has a total of 1457 frames averaging out to 131 frames per video. All the frames in the dataset have been annotated with 14 key-points i.e., head, neck, left-right shoulder, left-right elbow, left-right wrist, left-right hip, left-right knee and left-right ankle. In our experiments, we use only the upper body joints i.e., head, left-right shoulder, left-right elbow, left-right wrist.

These datasets vary in terms of activities, sampling rate, shape variance, and illumination. We demonstrate our experiments on a wide variety of datasets, which indicates the robustness of the proposed model.

4.6 Evaluation Measures

In all the experiments, we compare the estimated joints against frames with manual ground truth. We present results as graphs that plot accuracy vs normalized distance from ground truth, where a joint is deemed correctly located if it is within a set threshold distance from a marked joint centre in ground truth. Higher pck implies more accurate estimations.

4.7 Experiments and Results

4.7.1 Baselines

SFN. SFN [73] is a state-of-the-art ConvNet for human pose estimation. It is a fully convolutional network with an implicit spatial model that predicts a confidence heatmap for each body joint in images.

Figure 4.6 shows the architecture of SFN. It consists of a spatialNet (8 convolution layers) which regresses heatmap for each joint separately. The Spatial fusion layers take as an input pre-heatmap activations (conv3 and conv7), and learn dependencies between the human body parts locations represented by these activations. It learns to encode the dependencies between joints and learns an implicit spatial model to prune the kinematically impossible poses.

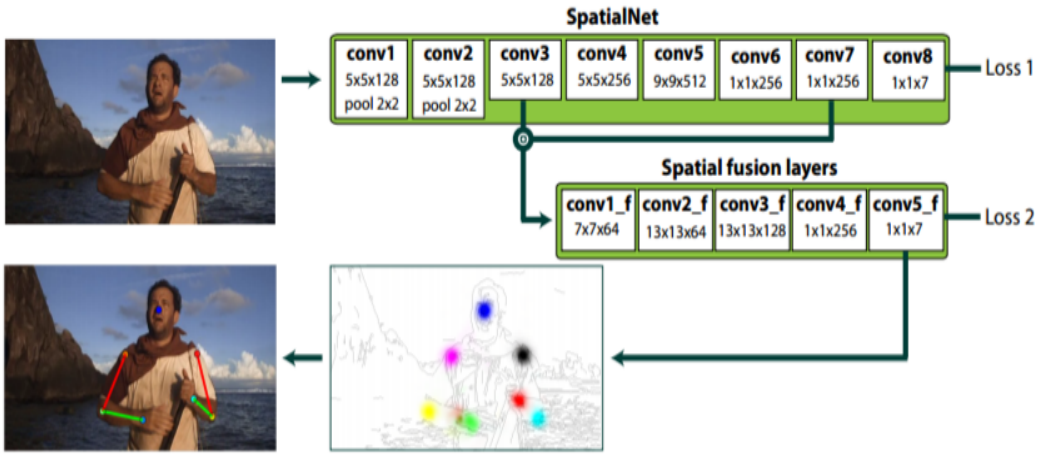


Figure 4.6: SFN. SpatialNet is a fully convolutional network. It regresses heatmap for each joint separately. The second part of network is Spatial Fusion Net, which takes intermediate activations from SpatialNet. The spatial fusion layers learn to encode dependencies between human body parts locations, learning an implicit spatial model.

YR. [95] is a method for detecting articulated people and estimating their pose from static images based on a new representation of deformable part models. The flexible mixture model jointly captures

spatial relations between part locations and co-occurrence relations between part mixtures, augmenting standard pictorial structure models that encode just spatial relations.

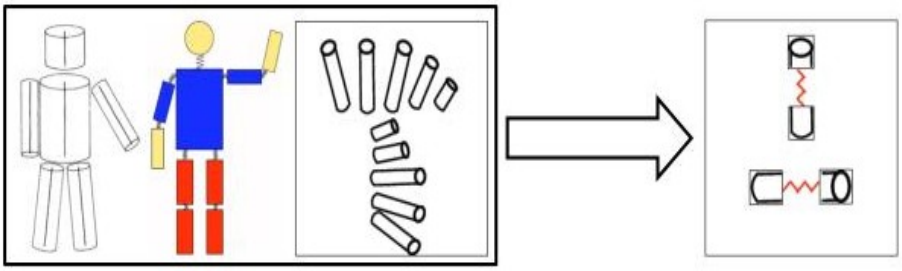


Figure 4.7: YR. First and second model, represent the classic articulated limb model of Marr and Nishihara [64] and Felzenszwalb and Huttenlocher [36] respectively. Third model, shows different orientation and foreshortening states of a limb, each of which is evaluated separately in classic articulated body models. On the right, Yang and Ramanan [95] approximates these transformations with a mixture of non-oriented pictorial structures, in this case tuned to represent near-vertical and near-horizontal limbs.

4.7.2 Training

The videos are split into fixed length sequences. To increase the total number of samples to train the model, we perform data augmentation. The frames are randomly rotated between -30° and 30° and only horizontally flipped. Data augmentation has to be done carefully so that the video generated after augmentation should be semantically meaningful. By considering overlapping sequences, there are two-fold advantages: (i) this increases the number of samples for training, and (ii) overlapping sequences generate multiple estimates for a single frame, which reduces the total error.

The data is split into mini-batches of size 64. The correction model is trained on the YouTube Pose dataset. We used Keras for our experiments. For experiments on CVIT-SPORTS, the correction model is fine-tuned on a subset of CVIT-SPORTS videos dataset. The model is trained for 100 epochs, using RMSProp optimizer. The learning rate is set to 0.01.

The YouTube Pose Subset accuracy (%) at $d = 20$ pixels is shown in Table 4.1. Our method surpasses SFN [73] by 2.07% . There is 5% boost in accuracy for head and shoulders improve by 6% (Table 4.1). Our method performs equal to the baseline on wrists and elbows. Table 4.1 shows that we surpass YR [95] by 3.7%. We see that our method corrects head, elbows and shoulders but doesn't improve on wrists (which are harder to define in complex poses). We show 18%, 1%, 3% boost over YR on head, elbows and shoulders respectively. Charles *et al.* [23] mentioned that YR model doesn't perform well on the YouTube Pose dataset. Hence, they have re-trained the model to improve the estimates. Hence, we see that the YR average pck is 44.0% (ref table 4.1) which is less compared to the accuracies mentioned in [23]. Experiments demonstrate that the proposed approach refines predictions given generic pose estimates.

Figure 4.8: Comparing human poses on sample images from YouTube Pose Subset and CVIT Sports videos dataset. (a), (b), (c), (d) show examples of pose corrections and (e), (f) show failure cases where red and yellow correspond to joints predicted using initialization (baseline) and our method respectively.

The PCK (Percentage of Correct Keypoints) accuracy on the CVIT-SPORTS videos dataset is shown in Table 4.2. We see improvement in head, elbows and shoulders over SFN. The head joint accuracy enhances by 26.2%. The accuracy averaged over all joints exceeds baseline by 3.1%. While using YR baseline, there is boost in wrists, elbows and shoulders and the average pck gain is 0.7%.

Figures 4.8 and 4.1 show the visualizations of pose corrections on sample dataset images. The red and yellow represent the initialization (baseline predictions) and the corrected pose predictions respectively. In Figure 4.8(a) and (b), the initial predictions for left elbow and left wrist are erroneous, but our approach corrects the poses as shown in the figure. Figure 4.8(c) predicts left wrist on the right wrist while Figure 4.8(d) predicts right wrist on the left wrist, and our method successfully corrects the pose. Our method fails to correct the poses, if the initial predictions are erroneous across the neighborhood. In Figures 4.8(e) and (f), it is not able to refine the pose well enough, as the neighborhood frames also have erroneous predictions, which makes it difficult for refinement. Adding to that, these videos have low sampling rate and large motion changes across frames. For example, Figure 4.8(f) is the only frame where the head position lies in center but its previous and next frames have head position in the top (as shown in Figure 4.8(a)) and this leads to the errors.

Method	Head	Wrsts	Elbws	Shldrs	Average
Pfister *et al.* [73]	74.4	59.0	70.7	82.7	71.3
SFN [73]	79.2	58.4	71.1	82.4	71.9
SFN$^{++}(Ours)$	**84.9**	56.8	71.0	**88.3**	**73.9**
YR [95]	44.6	30.3	37.9	64.1	44.0
YR$^{++}(Ours)$	**62.8**	29.4	**38.9**	**67.3**	**47.7**

Table 4.1: Component analysis on YouTube Subset Pose datasets. Accuracy (%) at d = 20 pixels. SFN^{++} and YR^{++} indicates refinement using the proposed method. (We have highlighted all results where the proposed method shows improvement.)

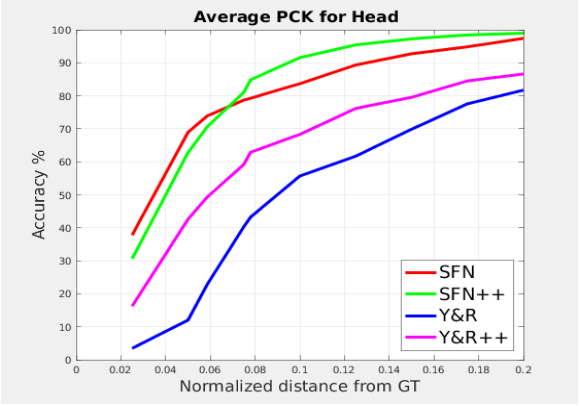
The PCK plots for head, wrists, elbows and shoulders on the YouTube Pose Subset dataset are shown in Figure 4.9. It is clear from the figure that the proposed approach has high recall. Also, the gain in accuracy is highest for head, followed by shoulders, elbows and wrists. Higher recall is an indication of refinement in joint predictions (Figure 4.1and 4.8).

4.8 Summary

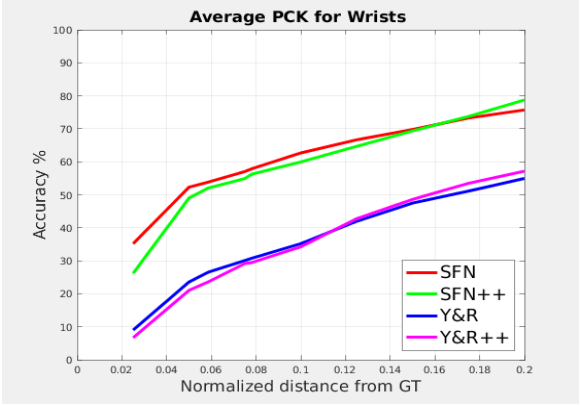
In this paper, we showed that the proposed pose correction model refines pose estimates obtained from generic models, independent of the pose estimator used to generate the initial pose estimates. We successfully posed the pose correction problem as a sequence-to-sequence learning problem. We demonstrated our results on challenging datasets which cover a wide range of activities, and are sampled at different sampling rates. The results show great promise in this approach to get more accurate pose estimation results in a simple, fast and generalizable manner.

Method	Head	Wrsts	Elbws	Shldrs	Average
SFN [73]	20.7	46.7	38.9	55.7	43.2
SFN$^{++}(Ours)$	**46.9**	42.7	**38.9**	**56.7**	**46.3**
YR [95]	78.9	43.9	49.8	73	59.1
YR$^{++}(Ours)$	78.6	**44.0**	**51.2**	**74.3**	**59.8**

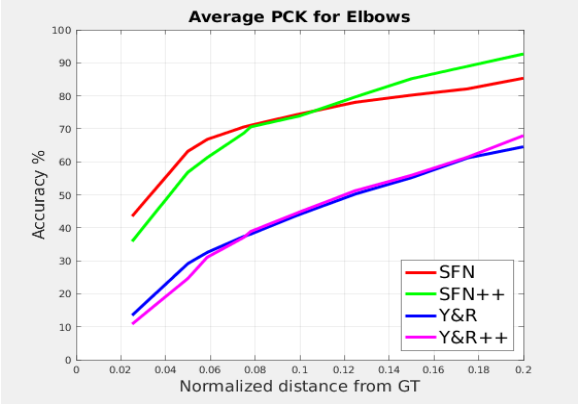
Table 4.2: Component analysis on CVIT SPORTS videos dataset. Accuracy (%) at d = 20 pixels. SFN^{++} and YR^{++} indicates refinement using the proposed method.(We have highlighted all results where the proposed method shows improvement.)



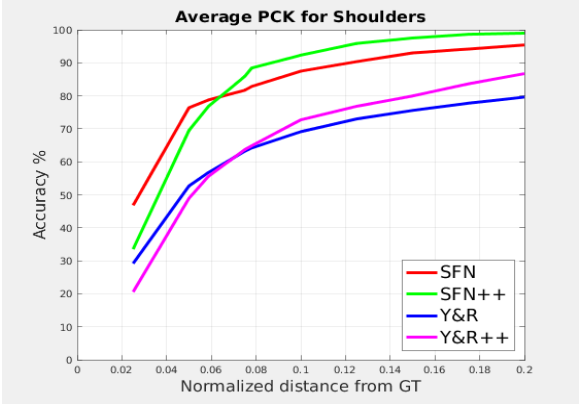
(a)



(b)



(c)



(d)

Figure 4.9: Results of our approach on YouTube Pose Subset dataset. We observe that the refined estimates using our approach have higher recall compared to the baselines: Yang & Ramanan [95] and Spatial Fusion Network [73].

Chapter 5

Conclusions and Future Work

With the success of deep learning and also to strive towards robust problem solving and benchmarking, it is essential to have large scale annotated data. It is evident that annotating huge amount of data manually is not feasible and prone to human error. There is a need for automation in data annotation, and there have been multiple attempts to address this task. Earlier works have either crowd-sourced the annotation task (which is expensive and not reliable) or have used vision and machine learning algorithms to develop annotation mechanisms.

In this thesis, we propose methods using vision and machine learning algorithms to automate the annotation task for multiple object annotations in video sequences. The aim is to increase awareness and prioritize building large datasets. We have shown that by using redundant information in videos, the human annotation efforts can be reduced significantly. The proposed solutions makes it feasible to generate large scale object annotations efficiently.

5.1 Future Work

Future research extensions of the proposed methods would be, application of present work for other object annotations (discussed in chapter 3) in video sequences.

- The method proposed in chapter 3, for tight bounding box annotation of objects can be directly extended for semantic labelling annotation of objects in video sequences.

- The annotation scheme for human pose in chapter 4, can be extended for hand pose annotation for hand gesture videos.

Apart from application of the proposed methods, future directions in terms of algorithm are

- In chapter 3, user annotates only the key-frames and the annotations are propagated across neighborhood. The key-frame selection strategy used in chapter 3, can be made more sophisticated based on the video content so that the user annotates minimum frames.

- Semi-automatic annotation scheme in chapter 3, requires human to annotate key-frames. This can be automated with the help of multiple algorithms for object bounding box detection, and select the most confident detection from set of predictions based on confidence scores for initialization.
- In chapter 4, the sequence to sequence learning formulation, is independent of the initialization scheme. By designing an end-to-end architecture and backpropagating the errors not only through the LSTM architecture but also till the initialization scheme, the initialization scheme itself can be improved.

We highlight the necessity of large scale annotated data specially in the era of deep learning. By proposing the automatic object annotation methods, we hope it would motivate the community in building larger datasets and not underrate the importance of data.

Related Publications

Conference

1. **Efficient object annotation for surveillance and automotive applications.**
   IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW), 2016
   Sirnam Swetha*, Anand Mishra* , Guruprasad M. Hegde+ , C.V. Jawahar*
   CVIT, KCIS, International Institute of Information Technology, Hyderabad*
   Bosch Research and Technology Centre, India+

2. **Sequence-to-Sequence Learning for Human Pose Correction in Videos**
   4th Asian Conference on Pattern Recognition (ACPR), 2017
   Sirnam Swetha* , Vineeth N Balasubramanian+ , C.V. Jawahar*
   CVIT, KCIS, International Institute of Information Technology, Hyderabad*
   Indian Institute of Information Technology, Hyderabad+

Bibliography

[1] http://web.mit.edu/vondrick/vatic/. 1, 18, 19

[2] http://viper-toolkit.sourceforge.net/. 18, 19

[3] http://www.research.ibm.com/videoannex/index.html. 18, 19

[4] G. Cheung A. Frome, A. Abdulkader, M. Zennaro, A. Bissacco B. Wu, and L. Vincent H. Adam, H. Neven. Large-scale privacy protection in street-level imagery. In *ICCV*, 2009. 1, 17

[5] 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. 9

[6] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008. 9, 10, 26

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009. 32

[8] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010. x, 2, 9, 10, 26

[9] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. x, 11, 13

[10] R. Arandjelović and A. Zisserman. Visual vocabulary with a semantic twist. In *Asian Conference on Computer Vision*, 2014. 12

[11] Mikel Ariz, José J. Bengoechea, Arantxa Villanueva, and Rafael Cabeza. A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. *Comput. Vis. Image Underst.*, 2016. 15

[12] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *Proceedings of the Second International Conference on Human Behavior Unterstanding*, 2011. 32

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, 2014. 32

[14] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940, 2013. 16

[15] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Trans. on Graphics (SIGGRAPH)*, 32(4), 2013. 12

[16] Simone Bianco, Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. An interactive tool for manual, semi-automatic and automatic video annotation. *CVIU*, 2015. 18, 20

[17] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, sep 2009. URL `http://www.eecs.berkeley.edu/~lbourdev/poselets`. 13

[18] Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, 2001. 22

[19] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *PAMI*, 2004. 22

[20] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008. 12

[21] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1513–1520, 2013. 16

[22] Vladimir Kolmogorov Carsten Rother and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 20, 22

[23] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Personalizing human video pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. x, 1, 4, 5, 13, 31, 36, 39

[24] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1, 32

[25] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 13

[26] Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 32

[27] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 11, 12

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. x, 1, 8, 9, 10

[29] David S. Doermann and David Mihalcik. Tools and techniques for video performance evaluation. In *ICPR*, 2000. 18, 19

[30] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 2012. 32

[31] Marcin Eichner and Vittorio Ferrari. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, 2009. 13, 32

[32] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2008. x, 4, 5, 10, 26

[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 9, 10, 12

[34] C. Thurau F. Nasse and G. A. Fink. Face detection using gpu-based convolutional neural networks. In *CAIP*, 2009. 1, 17

[35] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVPR Workshops*, 2007. 1, 18

[36] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 2005. xi, 32, 39

[37] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. 13

[38] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 12

[39] Wen Gao, Bo Cao, Shiguang Shan, Xilin Chen, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The cas-peal large-scale chinese face database and baseline evaluations. *Trans. Sys. Man Cyber. Part A*, 2008. 16

[40] Nicolas Gourier, Daniela Hall, and James L. Crowley. Estimating face orientation from robust detection of salient facial structures. In *FG NET WORKSHOP ON VISUAL OBSERVATION OF DEICTIC GESTURES*, 2004. 15

[41] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, 2013. 32, 36

[42] Mislav Grgic, Kresimir Delac, and Sonja Grgic. Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879, 2011. 16

[43] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 1, 18

[44] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 9, 10

[45] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 10

[46] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 16

[47] Arjun Jain, Jonathan Tompson, Yann LeCun, and Christoph Bregler. Modeep: A deep learning framework using motion features for human pose estimation. *CoRR*, 2014. 32

[48] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 10

[49] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar. Has my algorithm succeeded? an evaluator for human pose estimators. In *European Conference on Computer Vision*, 2012. 13

[50] N. Jammalamadaka, A. Zisserman, and C. V. Jawahar. Human pose search using deep poselets. In *International Conference on Automatic Face and Gesture Recognition*, 2015. 32

[51] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, 2013. 13

[52] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. x, 11, 13

[53] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. 2013. 32

[54] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 9, 32

[55] Andrzej Kasinski, Andrzej Florek, and Adam Schmidt. The put face database. *Image Processing and Communications*, 13(3-4):59–64, 2008. 16

[56] Isaak Kavasidis, Simone Palazzo, Roberto Di Salvo, Daniela Giordano, and Concetto Spampinato. An innovative web-based collaborative platform for video annotation. *Multimedia Tools Appl.*, 2014. 1, 18, 19

[57] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*. British Machine Vision Association, 2013. 13

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 1, 17

[59] M. Pawan Kumar, Philip H. S. Torr, and Andrew Zisserman. OBJCUT: efficient segmentation using top-down and bottom-up cues. *PAMI*, 2010. 20

[60] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692. Springer, 2012. 16

[61] Yann LeCun, Koray Kavukcuoglu, and Clment Farabet. Convolutional networks and applications in vision. In *ISCAS*, 2010. 18

[62] Xiaofei Li, Fabian Flohr, Yue Yang, Hui Xiong, Markus Braun, Shuyue Pan, Keqiang Li, and Dariu M Gavrila. A new benchmark for vision-based cyclist detection. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 1028–1033. IEEE, 2016. 10

[63] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. *Microsoft COCO: Common Objects in Context*. 2014. 9, 12

[64] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978. xi, 39

[65] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, 2001. 12

[66] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 1, 18

[67] Peter M. Roth Martin Koestinger, Paul Wohlhart and Horst Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 15, 16

[68] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 12

[69] Ara V. Nefian. Georgia tech face database. 10

[70] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 12

[71] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 10

[72] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 12

[73] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *IEEE International Conference on Computer Vision*, 2015. xii, 1, 31, 32, 34, 38, 39, 41, 42, 43

[74] Tomas Pfister, Karen Simonyan, James Charles, and Andrew Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Asian Conference on Computer Vision (ACCV)*, 2014. 1, 13, 32

[75] Deva Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems 19*. 2007. 32

[76] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017. 10

[77] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 12

[78] German Ros, Simon Stent, Pablo F Alcantarilla, and Tomoki Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *arXiv preprint arXiv:1604.01545*, 2016. 12

[79] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 1, 17

[80] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18, 2016. 16

[81] Benjamin Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *In Proc. CVPR*, 2013. 13

[82] Benjamin Sapp, Alexander Toshev, and Ben Taskar. *Cascaded Models for Articulated Pose Estimation*. 2010. 32

[83] Jurgen Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012. 1, 17

[84] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. 14

[85] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 50–58, 2015. 16

[86] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. x, 11, 12

[87] Digvijay Singh, Vineeth Balasubramanian, and CV Jawahar. Fine-tuning human pose estimations in videos. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, 2016. 13, 37

[88] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 18

[89] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014. 14

[90] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, August 2014. 14

[91] Jonathan J. Tompson, Arjun Jain, Yann Lecun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems 27*. 2014. 32

[92] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 32

[93] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 2013. 2, 18

[94] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 10

[95] Yi Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011. xi, xii, 31, 34, 38, 39, 41, 42, 43

[96] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. *arXiv preprint arXiv:1704.02612*, 2017. x, 14

[97] Jenny Yuen, Bryan C. Russell, Ce Liu, and Antonio Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, 2009. 18, 19

[98] Xenophon Zabulis, Thomas Sarmis, and Antonis A Argyros. 3d head pose estimation from multiple distant views. In *BMVC*, 2009. 15

[99] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016. 14

[100] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012. 15