Part Bricolage: Flow-Assisted Part-Based Graphs for Detecting Activities in Videos

Sukrit Shankar, Vijay Badrinarayanan, and Roberto Cipolla

Machine Intelligence Lab, Division of Information Processing,
University of Cambridge, UK

Abstract. Space-time detection of human activities in videos can significantly enhance visual search. To handle such tasks, while solely using low-level features has been found somewhat insufficient for complex datasets; mid-level features (like body parts) that are normally considered, are not robustly accounted for their inaccuracy. Moreover, the activity detection mechanisms do not constructively utilize the importance and trustworthiness of the features.

This paper addresses these problems and introduces a unified formulation for robustly detecting activities in videos. Our ***first contribution*** is the formulation of the detection task as an undirected node- and edge-weighted graphical structure called *Part Bricolage (PB)*, where the node weights represent the type of features along with their importance, and edge weights incorporate the probability of the features belonging to a known activity class, while also accounting for the trustworthiness of the features connecting the edge. Prize-Collecting-Steiner-Tree (PCST) problem [19] is solved for such a graph that gives the best connected subgraph comprising the activity of interest. Our ***second contribution*** is a novel technique for robust body part estimation, which uses two types of state-of-the-art pose detectors, and resolves the plausible detection ambiguities with pre-trained classifiers that predict the trustworthiness of the pose detectors. Our ***third contribution*** is the proposal of fusing the low-level descriptors with the mid-level ones, while maintaining the spatial structure between the features.

For a quantitative evaluation of the detection power of *PB*, we run *PB* on Hollywood and MSR-Actions datasets and outperform the state-of-the-art by a significant margin for various detection paradigms.

Keywords: Activity Understanding, Pose Estimation, Graph Structures.

1 Introduction

Recognition/classification of human activities in videos attempts to understand the movements of the human body using computer vision and machine learning techniques, and classify them in an already seen activity category/class. The evaluation of recognition procedures is generally done on the datasets where the videos are spatio-temporally cropped to the volume of activity. On the other hand, the activity detection task requires the correct classification of an activity along with its spatio-temporal localization. For practical applications, the detection task is more viable, and most activity detection techniques have employed an exhaustive sliding-window search methodology for this

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 586–601, 2014.

purpose. However, the sliding-window search based detection procedures are computationally very expensive. The recent work of [5] introduced a graph-based detection procedure which is computationally efficient, and can be made to incorporate various types of recognition procedures.

To handle recognition tasks, standalone low-level features like Histogram of Oriented Gradients (HoG) [7], Histogram of Optical Flows (HoF) [17] etc., although conventionally quite successful, have been lately found somewhat insufficient for complex datasets like Hollywood2 [23]. To improve performance, researchers have tried to build mid-level representations from these low-level features. With mid-level features, the recognition is based on the assumption that the pose detection/body part estimation is quite accurate, which limits the final accuracy. In cases where some flow information is used to do better estimation of poses, the possible conflicts owing to multiple and confusing body part detections are not resolved, resulting in the recognition of very limited types of activities. Similar problems have also proved to be an impediment to the accuracy of the state-of-the-art detection procedures.

Inspired by the work of [5], we formulate the activity detection as a graph problem, but introduce more generality in what the graph can represent. To show its significance, we propose novel techniques for extracting mid-level features in videos and fusing them with low-level descriptors. These techniques solve some of the major shortcomings of the state-of-the-art activity classification methods, and thus can also be used for the same under an appropriate binding framework.

1.1 Related Work and Problems

This subsection discusses the activity recognition/classification and detection approaches that have been adopted in recent times in the literature, highlighting their positive aspects along with the associated shortcomings. We delineate the low-level and mid-level feature representation based methods for activity classification, and also mention the major fallacies in the generality of the state-of-the-art activity detection frameworks[1]. Finally, we highlight our major contributions.

Low-Level Descriptors for Activity Classification: The most studied approaches thus far for activity recognition are based on the usage of low-level features with bag-of-words models. Introduced by [16], sparse space-time interest points and subsequent methods, such as local ternary patterns [41], joint sparse representations [12], dense interest points [37,30], better motion cues [14] and discriminative class-specific features [15], typically compute a bag-of-words representation out of local features and use them for classification. The work of [35] uses densely rather than sparsely sampled trajectories for better performance, and [36] builds upon this work to incorporate more types of low-level features while also accounting for camera motion. Fusing many low-level features with flow information can be looked upon as extracting abstract mid-level representations. [13] forms a mid-level representation using spatio-temporal patches

[1] Although this paper does not target the activity classification problem in isolation from the detection problem, we review the activity classification methods in order to convince the reader with the novelty of our proposed techniques of extracting mid-level features in videos with pre-trained classifiers that predict the trustworthiness of the part detectors, and fusing the low-level descriptors with the mid-level ones while maintaining the spatial structure between them.

consisting of object detections and low-level features. Their method is targeted more towards context based representation and less towards robustly modelling complex human movements. Some authors [26] have tried to somewhat extend the bag-of-words concept to form a high-level descriptor from a large number of small action detectors. The work of [44] follows a two-layered structured approach for activity classification, where the first layer encodes low-level features, and the second layer extracts mid-level representations called *Actons* from the first layer. Authors in [20] use a top-down approach where the top layer consists of coarse body parts, and the lower layers contain hierarchically segmented body portions. Their method however, uses low-level features for body-part estimation and hierarchical segmentation, and thus lacks robustness which limits their use for complex datasets.

Most of these methods are predominantly global recognition methods and are not well-suited for use in the recognition of complex activities; however, methods like [12,30,36,44] that have performed relatively well on complex datasets have indirectly built coarse mid-level representations from low-level features.

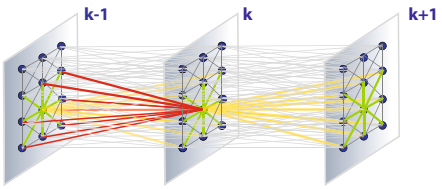

Fig. 1. This is the graphical structure of our Part Bricolage (PB) model discussed in Section 2.1. The figure shows all the possible connections between the nodes of three consecutive frames $k-1$, k, $k+1$ of a video. The *green* connections are highlighted to emphasize how a node is connected to its neighbours in the same frame; while the *yellow* connections indicate the connections of a node in frame k to the nodes in the next frame. The red lines show the connections to the node in frame k from nodes in frame $k-1$. The solution to the PCST problem over this graph finds an optimal subgraph that consist nodes and connections, representing the activity of interest. *Best Viewed in Color.*

Poses for Activity Classification: The work of [24] was one of the first to do recognition of basic activities by body part estimation. Many state-of-the-art pose estimation systems use action-specific priors to simplify the pose estimation problem [6,18,32]; while others use pose information for classifying actions [21,28,33,38]. Approaches using pose information for labelling activities mostly consider image datasets and depend on key-pose matching, while the methods using action specific priors for pose refinement typically require additional action labels. The key-pose matching techniques generally prove robust for very discriminative sequences and fail for complex datasets Apart from the requirement of additional training data, the technique of pose refinement from action labels suffers from the inability to account for occlusions. The work of [40] has tried to couple the two approaches, but the coupling is targeted more towards better 3D pose estimation for basic activities using flow information. The method does not tend to consider the ambiguities/conflicts that occur in real movie videos. The work of

[43] has proposed a 3D kinematics descriptor called the *Moving Pose*, but their method requires depth information for training and inference.

Some recent works for pose-based activity recognition have tried to use flow information with state-of-the-art pose detectors like poselets [2], flexible mixture of parts (FMP) [39] and stretchable models [27]. The work of [25] attempts to find key frames based on poselet detections for activity recognition. However, they do not consider the fact that poselets for complex videos can be conflicting and it is generally difficult to know the correct poselet based only on the probability scores[2]. As a result, they show very marginal improvements, even for datasets with basic interactive activities. Also, they expect the entire video dataset to be manually annotated with poselet bounding boxes, which is a serious limitation for video sequences. The work of [11] interleaves flow and pose information to cater to the inherent inaccuracy of the body part detectors. However, they do so only for lower arms, and their work is targeted more towards background-foreground segmentation in videos. Researchers in [34] estimate the body parts on a spatial and temporal basis using pose and flow information. However, their method tries to refine joints-based pose representations using motion fields, which is only robust for very simple actions like gestures, where joints can be estimated to a reasonable accuracy. Consequently, the method does not generalize to complex actions.

Activity Detection Methods: Template-based activity recognition methods attempt to do recognition by detection and therefore, can also be considered as activity detectors. The methods of [26,9,31,22] are the popular methods in this class. However, such methods do not form generic activity detection frameworks, since they are strictly tied to the underlying recognition procedures, and do not aim to do precise detection.

The work of [5] is the most recent approach that shows state-of-the-art results explicitly for activity detection. It considers the problem of precisely segmenting the spatio-temporal volumes of the desired activity by a max-weighted connected subgraph search (MWCS) methodology. Their approach is computationally efficient as compared to sliding-window search methods. However, they formulate the problem as a node-only-weighted graph, which limits the incorporation of the importance and trustworthiness of features, thereby lacking global generality.

We present a unified approach for activity detection, which addresses some of the key issues mentioned above. Our major contributions are as follows:

1. We formulate the task of detecting activities in unconstrained videos as an undirected node- and edge-weighted graphical structure called *Part Bricolage (PB)*, where the node weights represent the type of features along with their importance, and edge weights incorporate the probability of the features belonging to a known activity class, while also accounting for the trustworthiness of the features connecting the edge. Prize-Collecting-Steiner-Tree (PCST) problem [19] is solved for such a graph that gives the best connected subgraph comprising the activity of interest. Fig 1 provides an intuition of the graphical formulation.

[2] In the pose estimation systems such as poselets [2] and flexible mixture of parts (FMP) [39], although the probabilities of detections are also estimated alongside, they can only be trusted when the probabilities are high enough (say greater than 0.5). For lower probabilities (say less than 0.5), simply sorting them does not help to rank detections according to their degree of correctness. For complex videos, generally, a lot of detections are with low probabilities and thus, ambiguity resolving procedures demand an exploration.

2. We propose a novel technique for robust body part estimation, which uses two types of state-of-the-art pose detectors, and resolves the plausible detection ambiguities with pre-trained classifiers that predict the trustworthiness of the pose detectors.
3. We propose the fusion of low-level descriptors with the mid-level ones, while maintaining the spatial structure between them. This helps to better model the motion relationships in a video, specially when the detected mid-level features are sparse.

2 Approach

This section describes our *Part Bricolage (PB)* model, giving the necessary details and highlighting its various advantages. We first explain the complete formulation of our graphical structure, while also mentioning our technique of fusing mid-level features with the low-level ones (without compromising the spatial structure between them) for better motion representation. We then present our novel technique of robustly estimating the human body parts.

2.1 The Graphical Structure

Let us consider a video with K frames with each frame indexed as k ($k = 1, \ldots, K$). Let each frame in the video have N_k points (pixels to which features are associated) regularly spaced over the entire spatial region. As we shall later describe in the subsequent subsections, these points provide necessary information about the video. For consistency across frames , we make $N_k = N$, i.e. the number of points in each frame as same[3]. For such a video, we form a undirected graphical structure $G = (V, E, c, d)$ with node values $d : V \rightarrow \mathbb{R}^{\geq 0}$, edge costs $c : E \rightarrow \mathbb{R}^{\geq 0}$ and connections such that each node in a frame k is connected to its eight neighbouring points in frame k and nine neighbouring points in frame $k + 1$ (Fig 1). A node can possibly consist of multiple points; however, under a generic formulation, we consider each point as a node. Considering multiple points in a node reduces the granularity for doing detection.

We intend that for such a graph, the weights of the nodes should reflect the types of features that the nodes contain along with their importance, and the edge connections should contain the weights that indicate the cost of transiting from one node to another. If such a graph has to yield an activity of interest, then the edge costs should be more between the nodes that do not form a part of the targeted activity, and also between the nodes that represent the presence of features (related to human motion) with lower confidence levels. Where the two nodes which are connected together represent features relevant to the activity of interest and also with high confidence levels, the cost of transition between them should be less, indicating that the optimal subgraph that we wish to find out should contain such connected nodes.

Graph Optimization: We first describe the graph optimization problem that we intend to solve for localizing the activity of interest. Specifically, we find the solution

[3] For all our experiments, we take N equal to one-tenth the number of pixels in a frame. This choice is mostly empirical, and suffices when the activity of interest occupies most of the spatial region. For cases where the spatial occupation of the activities is less, N can be increased.

of a Prize-Collecting-Steiner-Tree (PCST) problem, which for an undirected, connected, node- and edge-weighted graph, finds the optimal subgraph maximizing the node weights and minimizing the edge costs. Given the way we define the node weights and the edge costs (as intuitively explained at the start of this subsection), solution to the PCST problem suffices for the activity detection task.

DEFINITION 1: (Max-Weighted-Connected-Subgraph (MWCS) Problem) - Given a connected, undirected, node-only-weighted graph $Z = (V_Z, E_Z, w)$ *with weights* $w : V_Z \rightarrow \mathbb{R}$*, find a connected subgraph* $T = (V_T, E_T)$ *of* $Z, V_T \subseteq V_Z, E_T \subseteq E_Z$*, that maximizes the score* $w(T) = \sum_{v \in V_T} w(v)$*.*

DEFINITION 2: (Prize-Collecting-Steiner-Tree (PCST) Problem) - Given a connected, undirected, node- and edge-weighted graph $G = (V_G, E_G, c, d)$ *with node values* $d : V_G \rightarrow \mathbb{R}^{\geq 0}$*, edge costs* $c : E_G \rightarrow \mathbb{R}^{\geq 0}$*, the (PCST) Problem [19] attempts to find a connected subgraph* $T = (V_T, E_T)$ *of* G*,* $V_T \subseteq V_G, E_T \subseteq E_G$*, that maximizes*

$$q(T) = \sum_{v \in V_T} d(v) - \sum_{e \in E_T} c(e) \tag{1}$$

We use the light-weight Heinz library provided by the authors of [8], which solves a Max-Weighted-Connected-Subgraph (MWCS) problem. Given a PCST problem over graph G ,we first convert it to a MWCS problem over an augmented graph Z, and then solve that using the Heinz library. We now show that such a conversion is theoretically feasible, and causes no alteration in the final solution.

PROCESS 1 - (Converting $G = (V_G, E_G, c, d)$ *to* $Z = (V_Z, E_Z, w)$*) - For every edge* $e \in E_G$ *connecting nodes* $u, v \in V_G$ *with edge cost* $c(e)$ *and node profits* $d(u)$ *and* $d(v)$*, form two edges* (u, a) *and* (a, v) *in* E_Z *by using an auxiliary node* $a \in V_Z$*, where* u *and* v *contain the same profits as in* G *(*$w(u) = d(u), w(v) = d(v)$*), and* $w(a) = -c(e)$*.*

The equivalence of G and Z easily follows from Definitions 1 & 2. Now, with the augmented graph Z, one must make sure that the optimal subgraph found by solving the MWCS problem over Z always contains the nodes u, v for an auxiliary node a, since the initial graph G never contained any a nodes. We thus state the following theorem:

THEOREM 1 - Given $G = (V_G, E_G, c, d)$ *with node values* $d : V_G \rightarrow \mathbb{R}^{\geq 0}$*, edge costs* $c : E_G \rightarrow \mathbb{R}^{\geq 0}$*, the vertex-weighted graph* Z *obtained by Process* 1*, when solved for the MWCS problem (Definition* 1*) can never contain a single-connected auxiliary node* $a \in V_Z$*.*

PROOF - For the graph $Z = (V_Z, E_Z, w)$ with weights $w : V_Z \rightarrow \mathbb{R}$, let the optimal Max-Weighted-Connected-Subgraph be $T = (V_T, E_T)$ of $Z, V_T \subseteq V_Z, E_T \subseteq E_Z$. Let $S \subseteq V_Z$ be a set of vertices, such that $\forall\ s \in S$, s is directly connected to $h \in V_T$. Let $S_{k'}; k' = 1, \ldots, K'$ represent all possible subsets of S. Then, since T is the optimal subgraph, $\forall\ k' = 1, \ldots, K'$

$$\sum_{h \in V_T} w(h) > \sum_{h \in V_T} w(h) + \sum_{s \in S_{k'}} w(s) \tag{2}$$

$$\Rightarrow w(s) < w(h) \tag{3}$$

Since, $w(a)$ is negative (Process 1), and weights of vertices u, v connected to a are positive (Definition 2),

$$w(a) < w(u), \quad w(a) < w(v) \tag{4}$$

Thus, if $a \in V_T$, neither of u or v can belong to $S_{k'}$ for any k', since that would contradict Equ (3) with Equ (4). But, if $a \in V_T$ is singly connected, atleast one of $S_{k'}$ should have either u or v. Hence, $a \in V_T$ cannot be singly connected. **This proves Theorem 1.** Note that in the above proof, we avoid the equality sign, assuming that the weights on vertices and edges of G are never zero. This is a valid assumption, since zero-weighted nodes and edges can always be deleted from the graph without affecting the cost of the optimal subgraph. It is important to note that for finding a solution to a PCST problem, all the node and the edge weights should be non-negative, i.e. $d : V \rightarrow \mathbb{R}^{\geq 0}$ and $c : E \rightarrow \mathbb{R}^{\geq 0}$.

From the above mathematical analysis, it is easy to see that one can convert the graph $Z = (V_Z, E_Z, w)$ to $G = (V_G, E_G, c, d)$ by assigning the negative of the minimum of the node weights of Z as the cost to all the edges in G, and adding the same to all the node weights of Z and assigning to G; provided that there is at least one node in Z with a negative weight. Then, solving PCST problem over G will be equivalent to solving the MWCS problem over Z. Thus, the two problems are related. However, if one formulates the activity detection task as a solution to the MWCS problem, it limits the design since all edge weights are same. In contrast, formulation of the detection task as a solution to the PCST problem offers flexible design choices.

Defining Node Weights: As stated earlier, we intend to represent mid-level (such as body parts) as well as low-level features (like optical flow [1]) by the nodes in the graph. We consider six body parts in a human - torso, head, two legs and two hands. Let $\boldsymbol{b_1}$ refer to the *bounding box* of a head, $\boldsymbol{b_2}$ that of a torso, $\boldsymbol{b_3}$ and $\boldsymbol{b_4}$ of two legs, and $\boldsymbol{b_5}$ and $\boldsymbol{b_6}$ of two legs. For a video frame, given human body-part detections, each point (node) on the frame can belong to one of $\boldsymbol{b_i}; i = 1, \ldots, 6$ or can be seen as not belonging to any of $\boldsymbol{b_i}$ (for nodes outside the human body parts). We define the weight of a node $v \in V$ of the graph G as follows:

$$d(v) = \left\{ 0.20i \; \forall \, v \subset \boldsymbol{b_i}, i = 1, 2 \;\; ; \;\; 0.60 \; \forall \, v \subset \boldsymbol{b_i}, i = 3, 4 \;\; ; \;\; 0.80 \; \forall \, v \subset \boldsymbol{b_i}, i = 5, 6 \right\} \tag{5}$$

If a point indicates a reasonable amount of flow field, but does not belong to any $\boldsymbol{b_i}$, it is assigned a weight of 0.5, else the point gets a weight of 0.01, indicating that it is not associated to any features under consideration. The node weights considered here define the importance of the features being considered. Note that for the nodes lying inside the bounding boxes of human body parts, we assign different weights based on the type of the body part that they represent. Since the human motions are more prominent due to legs and hands as compared to the head and torso, the nodes representing legs and hands are assigned higher weights. Since mid-level representations like poses are

more robust than mere flow information, the nodes representing only the flow information are assigned the middle weight, indicating that such information is less important than the detection of the limbs, but more than the detection of torsos and heads. In case a node happens to lie inside the bounding boxes of two features (due to partially overlapping bounding boxes), the node is made to represent the body part that would assign it a higher weight. For each node in the graph, let the probability of its occurrence be denoted by $p(v)$. For nodes not containing any of the body parts or the flow field, $p(v) = 0.01$. For the nodes containing the flow field, $p(v) = 1$, and for the nodes belonging to the body parts, $p(v)$ is assigned according to the trustworthiness of their detection as outlined in section 2.2.

Note above that we provide the flexibility of the body parts being represented in conjunction with the flow field. This maintains the spatial structure between the body parts and the flow information. Such a fusion offers us an advantage when the detections of the body parts are sparse. For instance, consider a case where the legs could not be detected within a frame, but the torsos and the head were detected. For a walking activity, the nodes representing the legs in the frame will be associated with a motion field, which when represented in a graphical structure naturally encodes motion relationships.

Before we discuss how the edge costs in our graphical structure are defined, we explain the training procedure. For videos in the training set, once the node weights are defined, we form a histogram over the entire video, one for each feature type (6 body parts and flow information). For each feature type, the bins represent the 6 body parts and 10 orientations of the flow descriptor, and the frequency of each bin indicates how many times the feature has occurred around a 50-frame temporal span of a node. Given the training videos and the associated activity class labels, binary linear-SVM classifiers using [4] are learnt for each each feature type. Thus, given a feature in the test set, once can predict whether the feature belongs to a known activity class or not, along with the degree of its presence. Note that during training, the probabilities of occurrences of the features and the edge costs are not considered. This is because, training is done on clean datasets with a pre-specified activity volume, and hence there lies no need to run a graph optimization problem. For test videos, the edge costs need to be incorporated according to the statistics of the training set and also the trustworthiness of the detected features.

Defining Edge Costs: The edge cost in the graph needs to be defined such that the cost is high if one is transiting to a node that represents a feature with lesser importance or lesser confidence level or the one which does not belong to an activity of interest. For an edge connecting any two nodes v_1 and v_2, if either of the nodes represent a feature that does not belong to any known activity class, the edge cost is assigned the maximum value of 1. In all other cases, the edge cost is defined as follows:

$$c(e_{(v_1,v_2)}) = \min(0.01, (|p(v_1) - p(v_2)|) \times (j(v_1) + j(v_2))/2) \tag{6}$$

where $p(v_1), p(v_2)$ are the probabilities of occurrences of the features at nodes v_1 and v_2 respectively, and $j(v_1), j(v_2)$ indicate the degrees to which the features at nodes v_1 and v_2 belong to a known activity class. A higher value of $j(v)$ indicates lesser presence of the feature in an activity class. All values are normalized so that the edge cost is always between 0 and 1. This is to prevent biasing in the graph.

Given a test video, once the node weights and the edge costs are assigned over the graph, we also store the activity class to which each node belongs. In case, the feature at a node does not belong to any known activity class, no information is stored. The PCST solution is computed over the graph, and the optimal subgraph is found representing the spatio-temporal localization of the activity. A histogram is computed over all nodes of this optimal subgraph, which indicates the number of nodes belonging to each known activity class. The class that exhibits the maximum frequency in the histogram is assigned to the test video.

2.2 Estimation of Human Body Parts

For a video with K frames, we start by running two state-of-the-art body part detectors, viz. poselets [2] and the flexible mixture of parts (FMP) [39], for frames separated by 0.25 sec in time duration. This is because, normally within this duration, poses in an activity do not change significantly enough that they cannot be tracked with the flow information. This condition mostly suffices for sports videos as well. Choosing a sparse set of frames for pose detection not only reduces the computational complexity, but also relaxes the requirement of a highly accurate pose detector. Let there be $m = 1, \ldots, M; M < K$ frames for which we run body part detectors.

Learning Classifiers: The FMP and poselet detections are not accurate for all types of poses. Although, they both return a probability score that indicates the accuracy of the detection, we observe that for lower probability scores, the detections cannot be ranked according to the degree of their accuracy by simply sorting these scores. We therefore, try to learn classifiers for both the poselets and FMP, which can indicate the trust in the detection scores. For this, we form an image dataset consisting of images from PASCAL VOC 2007 [10] and INRIA and Buffy image datasets considered in [39]. This dataset consists of around 1100 images with full body poses, partial body poses, multiple and overlapping poses, and null poses.

For FMP annotation, we run FMPs on each of the images of our image dataset, note the returned probability scores and the body-part detections, and manually annotate whether (a) the detection was fully accurate (all 6 body parts were correct) - *category* C_1, (b) the detection was correct for head and torso, but was erroneous for some/all limbs - *category* C_2 and (c) the detection was not acceptable (no more than 1 out of 6 body parts were correct) - *category* C_3. We have thus three categories and the associated probability scores. Using this, we learn linear-SVM classifiers [3] using the LIBSVM library [4], which given a probability score categorizes the FMP detection. Note that for detections with FMP, we utilize the code provided by the authors of [39], and use their pre-trained model. The accuracy of the classifiers of these classes is evaluated by doing 50 random initializations of training and test data set (with a 50% train/test split). We always achieved the classification accuracy of around 90% for the test dataset. This shows that the classifiers that we have learnt from manual annotation can be trusted[4].

[4] Note that we learn classifiers by annotating a dataset of images, and not the video datasets under consideration. These classifiers are learnt once and need not be changed depending on the video dataset used for evaluation.

Fig. 2. First Row - *(Left Column)* The figure shows the torso and human body detections using our adaptive threshold with the poselets, without which no parts were detected. *(Right Column)* The figure shows some accurate FMP detections (belonging to the class C_1). **Second Row -** *(Left Column)* The figure shows FMP detections for the class C_2 where the head and torsos can be trusted, but not the limbs. *(Right Column)* The figure shows some FMP detections belonging to the class C_3. In such a case, the FMP detections cannot be trusted at all. As a result, our algorithm then depends solely on the torso and head detections from the poselets. **Third Row -** The figure shows the torso and human body detections using poselets, where the FMP detections belonged to the class C_3. **Fourth Row** - The figures show multiple torso and head detections using poselets. Using the approach specified in Algorithm 1, the correct torso detections were found out. *Best Viewed in Color.*

Poselets have a major advantage of predicting the viewpoints of body parts as compared to FMPs. However, since we do not model viewpoints in our framework, and the number of masks associated with the limbs in the poselets are comparatively much lesser than those of torso and head; we use poselet estimations only for the torso and full human body detection. For poselets, we perform detections using the code provided by [2]. We observe that the detection threshold (the probability score above which the torso detections and human detections are considered valid) set in the code of [2] many a times misses some key torso/poselet detections. Thus, we make the detection threshold for poselets adaptive in nature, i.e. we consider all poselet firings until *atleast* two torsos are detected in an image. The threshold can also be adapted so as to make more than a minimum of two torso detections, but we choose only two, since we do not have collective activities in our video datasets. We then note the returned probability scores for various torso and human body firings with poselets, note their regions of

Algorithm 1: Choosing Appropriate Body Parts

foreach *frame* m *discover the best part detections using learnt classifiers for FMP* **do**

(a) Initialize all nodes v with weights $d(v) = 0.01$ (from Eqn (5)) and probability of occurrence $p(v) = 0.01$. Initially none of the nodes belong to any body part.

(b) Run FMP and identify the category C of detection.

if $(C == C_1)$ *correct detection* **then**

(c) Include the FMP detected parts for the frame.

(d) Assign the probability of 1 to the nodes contained inside each detected part. **else if** $(C == C_2)$ *limbs may be missing but torso and head can be trusted* **then**

(e) Include the detected torso and the head for the frame.

(f) Assign probability of 1 to the respective nodes.

(g) Include the detected limbs for the frame.

(h) Assign probability of 0.5 to the respective nodes. **else** $C = C_3$ *& FMP detection cannot be trusted*

(i) Discard the FMP detections.

end

foreach *frame* m *detect torsos and human bodies using poselets* **do**

(j) Run Poselets and note the torso and human body detections

if *multiple torsos are contained inside a human body box* **then**

(k) Associate the human body box to the torso having the highest detection score.

end

if *a single torso is contained inside multiple human body boxes* **then**

(l) Associate the torso to the human body box to which it is most symmetrical.

end

One now establishes one-to-one mapping between a torso and a human body detection

foreach *torso-human body pair* **do**

(m) Estimate the head part of the body.

end

foreach *torso-head pair* **do**

(n) Check if the torso and head have a significant overlap with any of FMP detections

if *significant overlap occurs* **then**

(o) Discard the torso and head detections and continue with the FMP ones. **else**

(p) Consider the torso and the head detection.

(q) Assign probability of 1 to the respective nodes.

end

end

Apply Algorithm 2

detections, and manually annotate the images into the following two categories: *category* C_4 - where torso detections (with poselets) are not correct, and *category* C_5 - where torso detections (with poselets) are correct. We observe that segregating the torso and human body detections with poselets based on such probability scores in not feasible, since there are generally many good detections even with very low probabilities. Thus, no classifiers are learnt for the same. However, a higher probability score generally indicates a more confident detection. Also, a torso that is more symmetrically placed within the human body generally indicates a better detection.

After running the poselets and FMPs on a given video, we get many part detections along with their probabilities of occurrences for each of the M frames. We select the most appropriate parts amongst them using Algorithm 1, for which we reuse the learnt classifiers (mentioned above). For predicting the spatial position of the body parts between m^{th} and $(m+1)^{th}$ frames, we utilize the flow information (see Algorithm 2). We reiterate that for any frame where part detections are included according to Algorithm 1, the overlapping of the bounding boxes of the parts is not a problem, since a node is always assigned to the bounding box of the part that gives it the maximum weight (as discussed in Section 2.1 - *Defining Node Weights*). See Fig 2 for getting a pictorial representation of ideas presented in this sub-section and Algorithms 1 and 2.

Algorithm 2: Estimate body parts for the frames between m and $m+1$

m and $m+1$ do not represent adjacent frames, but those for which detection is done one after the other

foreach *frames (m, m + 1)* **do**

  foreach *bounding box of the body part detected in m* **do**

    Track the body part of frame m in $m+1$

    (T1) For the bounding box of a body part in m, $\boldsymbol{b_m} \in \boldsymbol{b_i}$, find the flow field using optical flow [1] between frames m and $m+1$. Let $Y(\boldsymbol{b_m})$ refer to the type of body part that $\boldsymbol{b_m}$ contains (torso, hands, etc.). Use flow field to estimate the bounding box of that body part in frame $m+1$ as $\boldsymbol{b_{m+1}}$. Let p_{b_m} refer to the probability of occurrence of $Y(\boldsymbol{b_m})$, and $p_{b_{m+1}}$ to that of $Y(\boldsymbol{b_{m+1}})$. Initialize $p_{b_{m+1}} = p_{b_m}$.

    (T2) If frame $m+1$ contains body parts of type $Y(\boldsymbol{b_m})$, find the spatial locations of all such parts. If any such body part with bounding box $\boldsymbol{b_p}$ and probability of occurrence p_p is in a close neighbourhood (typically one-tenth of the size of the frame) of $\boldsymbol{b_{m+1}}$, then $\boldsymbol{b_{m+1}} = \boldsymbol{b_p}$ and $p_{b_{m+1}} = p_p$. In case of multiple parts near the neighbourhood of $\boldsymbol{b_{m+1}}$, the closest one is considered.

    (T3) Let H_1 be the color histogram of the part in $\boldsymbol{b_m}$ and H_2 for that in $\boldsymbol{b_m + 1}$. Compute the mass in the difference between H_1 and H_2 and divide it by the the mass in H_1, to give r_{c_H}. This value gives the change in appearance of the part.

    Estimate the position and probability of the occurrence of that part for the frames in between

    (T4) Penalize $p_{b_{m+1}}$ for the difference in appearance. So, $p_{b_{m+1}} := p_{b_{m+1}}(1 - r_{c_H})$

    (T5) For a frame n in between m and $m+1$, $\boldsymbol{b_n}$ is found by linearly interpolating $\boldsymbol{b_m}$ and $\boldsymbol{b_{m+1}}$. Similarly, p_{b_n} is found by linearly interpolating p_{b_m} and $p_{b_{m+1}}$.

  end

end

3 Results and Discussion

This section presents the results obtained with our *Part Bricolage* model for the task of activity detection. Note that for activity detection, the task is to predict the spatio-temporal bound of an activity along with the class label.

Table 1. Activity Detection Results with Part Bricolage (PB): Mean Overlap Accuracy for temporal detection on Hollywood (AP = AnswerPhone, GC = GetOutCar, HS = HandShake) dataset; and temporal and spatio-temporal detection (using full-person ground truth) on MSR-Actions dataset. It can be seen that the full *PB* outperforms the state-of-the-art procedures by a significant margin for all detection paradigms. The results deteriorate significantly for the Hollywood dataset if we do not use poselet detections. This is because the Hollywood dataset contains many partial body poses, where FMPs do not work well. Also, when the flow information is not fused with the mid-level body part detections, the accuracy gets affected, thereby justifying our design choice. *Best Viewed in Color.*

	Hollywood (Temporal)								MSR (Temporal)			MSR (Spatio-Temporal)		
	AP	GC	HS	Hug	Kiss	SitDown	SitUp	StandUp	Box	Clap	Wave	Box	Clap	Wave
ST-SubVol [42]	0.29	0.22	0.33	0.44	0.42	0.28	0.20	0.30	0.07	0.06	0.26	0.045	0.017	0.101
MWCS [5]	0.39	0.29	0.41	0.52	0.49	0.37	0.38	0.37	0.09	0.17	0.29	0.047	0.063	0.112
PB (Ours) - Poselets	0.21	0.18	0.31	0.29	0.29	0.27	0.24	0.31	0.19	0.25	0.39	0.131	0.114	0.201
PB (Ours) - Flow	0.35	0.29	0.41	0.42	0.39	0.35	0.32	0.36	0.11	0.18	0.31	0.066	0.078	0.165
PB (Ours) Full	**0.45**	**0.36**	**0.49**	**0.56**	**0.50**	**0.46**	**0.41**	**0.46**	**0.21**	**0.26**	**0.43**	**0.147**	**0.127**	**0.235**

We use the uncropped Hollywood [17][5] and the MSR-Actions [42] datasets for the evaluation of our *PB* model for activity detection purposes. The Hollywood dataset can be considered as a subset of Hollywood2 dataset, and contains around 470 videos having 8 action classes, viz. *AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp*. For detection on Hollywood dataset, we train with the cropped clips, and test with the uncropped videos. The train/test split is around 50% and the videos are chosen as specified in [17]. The MSR-Actions dataset is quite different from the Hollywood dataset, since the test sequences normally contain multiple actions with people frequently crossing each other and changing their position over time. Thus, MSR-Actions dataset presents a very good validation benchmark for the activities with dynamic occlusions. The dataset contains 16 videos having 3 action classes, viz. *Boxing, Hand Clapping, Hand Waving*. Since the KTH dataset [29] also contains these three action classes, we train using the KTH videos and test on all the sequences of the MSR-Actions dataset. This is a standard norm for activity detection as recommended by [42,5].

Our *Part Bricolage* model is specifically targeted for activity detection. Although the idea of robust body part estimations, and fusing of low-level and mid-level features in a graph can be utilized for activity classification as well, a seemingly different binding framework may be more suited. The graphical structure based binding framework that we have adopted in this paper is best suited for the problem of activity detection. To evaluate *PB* for detection, we thus choose datasets which contain some real movie activities (like in Hollywood) or simple activities with a lot of dynamic occlusions (like in MSR-Actions dataset), where detection task is challenging. Also, the state-of-the-art activity classification procedures cannot be directly incorporated in either of our PCST-type framework or MWCS-framework [5], since those procedures neither possess an inherent quantization of their descriptors, nor any notion of trustworthiness of features.

[5] Hollywood dataset contains both the noisy *uncropped* versions of the video sequences which contain about 40% extraneous frames, as well as the *clean* or cropped versions of the sequences, which have been trimmed temporally to the action of interest.

We present temporal detection results on the Hollywood dataset, and the temporal and spatio-temporal detection results on the MSR-Actions dataset. Note that since the activities in Hollywood dataset are spatially trimmed, only temporal detection is required. Table 1 presents all the detection results. We use the mean overlap accuracy as the evaluation metric, following [42,5]. For both temporal or full spatio-temporal detection, this metric computes the intersection of the predicted detection region with the ground truth, divided by the union.

Table 1 presents the detection results with our *PB* model, while also showing some intermediate results in order to justify our design choices. It is clear that *PB* model outperforms the state-of-the-art procedures by a significant margin on both the datasets, for the temporal as well as the spatio-temporal detection paradigm. It can be seen that results without the incorporation of the part detections from poselets deteriorate significantly for the Hollywood dataset, unlike the MSR-Actions dataset. This is understandable since the MSR-Actions dataset generally contains full body poses for which FMP part detections are quite accurate. This is not the case for the Hollywood dataset, where one finds lesser number of full body poses, and FMP part detections mostly fail. It can also be seen that the results without the incorporation of any flow information (low-level feature) shows deterioration for both the datasets. This clearly establishes the advantage of our proposal of fusing the mid-level features like body parts with the low-level features like optical flow within the graph structure. The lack of flow information affects the detection accuracy for the videos, where the mid-level representations are sparse, or complete body poses cannot be estimated with acceptable trustworthiness and the missing part show some movement. Also, in the cases where there are a lot of dynamic occlusions, flow information helps to separate the activity of interest.

It is noteworthy that comparisons in Table 1 are made after making the temporal and spatial granularity of our *PB* model to 5 frames (instead of one frame) as done in [5]. We consider a block of 25 points (5 points per frame for 5 frames) as a node here. This is consistent with our discussion in Section 2.1 where we mentioned that a node can possibly contain many points to decrease the granularity.

4 Conclusions and Future Work

We have introduced a unified formulation for robustly detecting activities in videos. Central to our formulation is an undirected node- and edge-weighted graphical structure called *Part Bricolage (PB)*, where the node weights represent the type of features along with their importance, and edge weights incorporate the probability of the features belonging to a known activity class, while also accounting for the trustworthiness of the features connecting the edge. Prize-Collecting-Steiner-Tree (PCST) problem [19] is solved for such a graph that gives the best connected subgraph comprising the activity of interest. We have introduced a novel technique for robust body part estimation, which uses two types of state-of-the-art pose detectors, and resolves the plausible detection ambiguities with pre-trained classifiers that predict the trustworthiness of the pose detectors. We have also proposed the fusion of low-level descriptors with the mid-level ones, while maintaining the spatial structure between them. Quantitative results establish the advantages of our various design choices, and show that our *PB* model outperforms the state-of-the-art detection procedures by a significant margin.

PB model can be extended to have a human-detector-initiated graph partitioning which can cater to simultaneous activities. Also, the distinction between left and right limbs can be made explicit. Better parametric models can also be incorporated, and node and edge weights of the graph can be associated with probabilistic graphical frameworks to detect collective, highly contextual as well as subtle human motions.

Acknowledgements. We thank Dr. Gunnar W. Klau and Mohammed El-Kebir of CWI (Centrum Wiskunde & Informatica) Life Sciences Group at Amsterdam, Netherlands for providing us the Heinz library, and for giving us the directions to solve the PCST problem using it.

References

1. Black, M.J., Anandan, P.: A framework for the robust estimation of optical flow. In: Proceedings of the Fourth International Conference on Computer Vision, pp. 231–236. IEEE (1993)
2. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV (2009)
3. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
4. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Chen, C.Y., Grauman, K.: Efficient activity detection with max-subgraph search. In: CVPR (2012)
6. Chen, J., Kim, M., Wang, Y., Ji, Q.: Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In: CVPR (2009)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
8. Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Müller, T.: Identifying functional modules in protein–protein interaction networks: an integrated exact approach. Bioinformatics 24(13), i223–i231 (2008)
9. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: CVPR (2003)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007), `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`
11. Fragkiadaki, K., Hu, H., Shi, J.: Pose from flow and flow from pose. In: CVPR (2013)
12. Gopalan, R.: Joint sparsity-based representation and analysis of unconstrained activities. In: CVPR (2013)
13. Jain, A., Gupta, A., Rodriguez, M., Davis, L.S.: Representing videos using mid-level discriminative patches. In: CVPR (2013)
14. Jain, M., Jégou, H., Bouthemy, P., et al.: Better exploiting motion for better action recognition. In: CVPR (2013)
15. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR (2010)
16. Laptev, I.: On space-time interest points. IJCV 64(2-3), 107–123 (2005)
17. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)

18. Lee, C.S., Elgammal, A.: Coupled visual and kinematic manifold models for tracking. IJCV 87(1-2), 118–139 (2010)
19. Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G.W., Mutzel, P., Fischetti, M.: An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. Mathematical Programming 105(2-3), 427–449 (2006)
20. Ma, S., Zhang, J., Ikizler-Cinbis, N., Sclaroff, S.: Action recognition and localization by hierarchical space-time segments. In: ICCV (2013)
21. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: CVPR (2011)
22. Malgireddy, M., Inwogu, I., Govindaraju, V.: A temporal bayesian model for classifying, detecting and localizing activities in video sequences. In: CVPR (2012)
23. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
24. Ramanan, D., Forsyth, D.A.: Automatic annotation of everyday movements. In: NIPS (2003)
25. Raptis, M., Sigal, L.: Poselet key-framing: A model for human activity recognition. In: CVPR (2013)
26. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR (2012)
27. Sapp, B., Weiss, D., Taskar, B.: Parsing human motion with stretchable models. In: CVPR (2011)
28. Schindler, K., Van Gool, L.: Action snippets: How many frames does human action recognition require? In: CVPR (2008)
29. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: ICPR (2004)
30. Shi, F., Petriu, E., Laganiere, R.: Sampling strategies for real-time action recognition. In: CVPR (2013)
31. Sullivan, M., Shah, M.: Action mach: Maximum average correlation height filter for action recognition. In: CVPR (2008)
32. Taylor, G.W., Sigal, L., Fleet, D.J., Hinton, G.E.: Dynamical binary latent variable models for 3d human pose tracking. In: CVPR (2010)
33. Thurau, C., Hlavác, V.: Pose primitive based human action recognition in videos or still images. In: CVPR (2008)
34. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: CVPR (2013)
35. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
36. Wang, H., Schmid, C., et al.: Action recognition with improved trajectories. In: ICCV (2013)
37. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., et al.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
38. Yang, W., Wang, Y., Mori, G.: Recognizing human actions from still images with latent poses. In: CVPR (2010)
39. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
40. Yao, A., Gall, J., Van Gool, L.: Coupled action recognition and pose estimation from multiple views. IJCV 100(1), 16–37 (2012)
41. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: ICCV (2009)
42. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
43. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection. In: ICCV (2013)
44. Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with actons. In: ICCV (2013)