Human Pose Estimation in Videos using Biposelets

Sam Toyer

Australian National University, Canberra, Australia
u5568237@anu.edu.au

Abstract. The problem of human pose estimation in still images has been well-studied in recent years, but making effective use of the temporal information inherent in videos is still an open problem. This paper presents a new model which is forced to learn temporal relationships by predicting poses in several frames at a time. The new approach caters to the detection and classification capabilities of convolutional networks by casting pose estimation as a problem of detecting the *biposelets* which constitute a pair of poses in adjacent video frames. Relative to existing single-frame-at-a-time methods, this new approach also makes it simpler to combine pose predictions into a coherent sequence of poses across an entire video. Experiments show that a biposelet approach outperforms previous work on shoulder localisation, but that localisation of wrists remains a challenge.

1 Introduction

Human pose estimation is the task of localising the joints of a person in an image or throughout a video sequence. Effective pose estimation must accommodate motion blur and self-occlusion, be invariant to subject clothing and scene clutter, and exhibit an understanding of anatomic constraints on poses; these requirements make pose estimation a challenging problem even with state-of-the-art computer vision techniques. In practice, 2D pose estimation is useful as a preprocessing step for higher-level tasks, including 3D pose estimation [1,2] and action recognition [3,4].

Intuitively, pose estimation in videos ought to be more effective than pose estimation in static images. Even if a pose is blurred or occluded in one frame of a video, it is often possible for humans to approximate it well by making use of the context provided by surrounding frames. Algorithmically leveraging this temporal redundancy is challenging. Simply feeding "temporal features" like optical flow or multiple frames to a static pose estimator results in only modest accuracy improvements [5]. On the other hand, extending a model to explicitly consider joint motion can yield intractable inference problems which require complex approximations [6]. Finding new ways to exploit temporal information is thus an active area of research.

The main contribution of this paper is to propose and test a new approach to human pose estimation in videos based on the concept of "biposelets". As

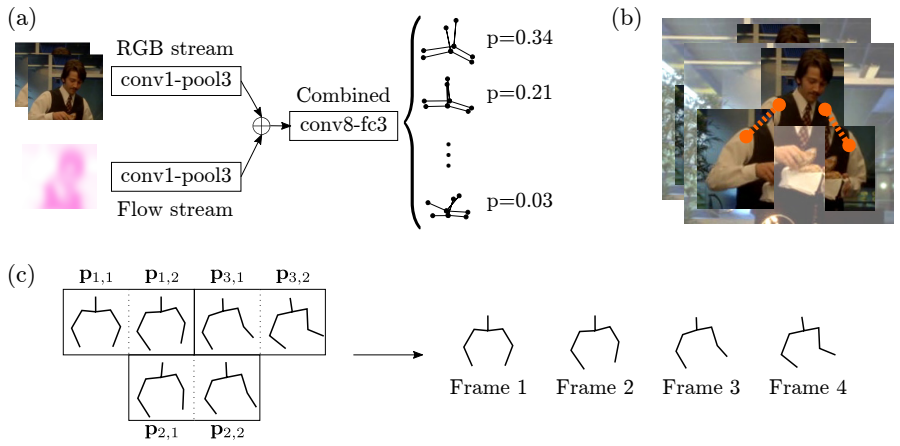

Fig. 1. An illustration of the pipeline. (a) depicts the biposelet-classifying CNN (Section 3.2), (b) depicts the graphical model used for subpose localisation (Section 3.3), and (c) depicts the stitching process (Section 4).

explained in Section 3.1, a biposelet describes a configuration of some subset of a person's joints in two adjacent video frames together; this allows biposelets to express both the position and instantaneous motion of joints. Hence, casting pose estimation as biposelet detection problem forces the proposed model to learn to make use of the information present in two frames of video at the same time. As a bonus, this pairwise detection approach gives rise to a relatively simple notion of temporal consistency of poses across a video sequence, which is elucidated in Section 4.

The complete pose estimation pipeline is divided into two stages, both depicted in Figure 1 (a–c): in the first stage (a–b), each pair of frames is processed independently of the others to produce sets of candidate pose pairs. The first stage makes use of a Convolutional Neural Network (CNN) to find regions of each frame pair which visually resemble different biposelets (a), followed by a graphical model which ensures that the relative positions of predicted biposelets are anatomically reasonable (b). In the second "stitching" stage (c), a single pair of poses is chosen from each set of candidate pose pairs; the resultant sequence of pose pairs can then be turned into a sequence of single poses by averaging the two poses predicted for each frame.

In Section 6, this strategy is evaluated on three established pose estimation benchmarks. The evaluation shows a significant improvement in shoulder localisation accuracy compared to existing work.

2 Related work

Pose estimation methods for static images typically build on some sort of appearance model which can be used to determine whether a specific joint is present in

a small image patch. Linear classifiers applied to histogram-of-gradients features are commonly used for this purpose [7,6], but have been eclipsed by increasingly sophisticated convolutional network architectures [5,8,9,10,11]. The choice of a CNN-based model in this work reflects this trend.

Appearance models can be complemented by global constraints on the relative positions of joints. For instance, Yang and Ramanan [7] define a graphical model which encourages joint locations to fall in regions of the image with high appearance scores while penalising atypical limb lengths and orientations. Several approaches also use the appearance of body parts to infer "types" which characterise those parts' positions relative to their neighbours [7,9]. The approach presented here is similar, although it differs from past work in attempting to simultaneously localise entire subsets of a person's joints instead of a single joint at a time.

In videos, pose estimation is frequently accomplished using a tracking-by-detection approach: first, a set of poses is estimated independently for each frame of the video. Next, the estimates for each frame are combined into a single temporally coherent sequence [1,12,6,13]. This work follows a similar pattern, but applies a detector to pairs of frames instead of a single frame at a time. As mentioned previously, this makes it simple to stitch predictions into a complete sequence. In contrast, past work using the frame-at-a-time method has had to resort to more complex flow- and appearance-based heuristics to ensure temporal consistency between predicted poses [6].

There have been several attempts to exploit motion information at the CNN level rather than just through stitching heuristics: MoDeep [5] simply augmented a single-frame heatmap regressor for joint positions with "motion features" derived from neighbouring frames. For localisation of fast-moving joints, the authors reported a boost in performance over an equivalent model without motion features, but little boost in performance for slower-moving joints like the elbows. Their result suggests that temporal relationships between joint position were not being learnt effectively.

Other approaches to learning temporal relationships at the network level include "flowing convnets" [10], and the recurrent network approach of Fragkiadaki et al. [14]. Flowing convnets use optical flow to warp joint position heatmaps between frames, then pool the heatmaps at each frame to improve accuracy; despite yielding state-of-the-art accuracy, flowing convnets only "understand" temporal relationships insofar as they are able to learn a small set of pooling weights for combining backwarped heatmaps. In contrast, Fragkiadaki et al. attempt to learn temporal relationships directly using a recurrent neural network architecture. Both approaches are benchmarked in Section 6.

3 Detecting pose pairs

3.1 Subposes and biposelets

The first stage of the pose estimation pipeline inspects two video frames at a time and infers a set of poses which may be present in each. For the purposes

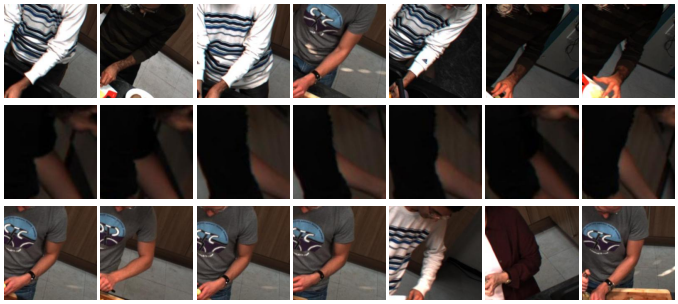

Fig. 2. Biposelets learnt for left elbows in the MPII Cooking Activities dataset. Cells within a row show different instances of the same biposelet. For brevity, only the first frame associated with each biposelet is shown.

of this stage, a pose is decomposed into a fixed set of subposes, each of which correspond to a subset of joints in the original pose. Subposes are chosen so that each subpose shares exactly one joint with neighbouring subposes, and so that each joint in the original pose is present in at least one subpose. For instance, a pose containing shoulder, elbow and wrist joints could be decomposed into one subpose containing the left wrist and left elbow, one containing the left elbow and left shoulder, one containing both the left and right shoulders, etc.

Rather than representing the locations of each joint in each subpose directly, the first stage discretises the space of configurations for joints within each subpose into K *biposelets*. Biposelets are so named in analogy to the *poselets* of Bourdev and Malik [15]. Unlike their namesakes, however, biposelets define positions of joints in two frames rather than just one. This allows biposelets to represent both the relative positions of joints and their movement over time. A representative set of biposelets learnt for the MPII Cooking Activities pose estimation dataset [16] is shown in Figure 2.

3.2 Frame pair model

Formally, the frame pair model conceptualises a pose as a tree-structured graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ consisting of a set of subposes $\mathcal{S}$ and a set of edges $\mathcal{E} \subset \mathcal{S} \times \mathcal{S}$. Subposes are chosen according to the constraints listed in Section 3.1, and a pair of subposes (s_1, s_2) will have an edge between them iff they have a joint in common. Part (b) of Figure 1 illustrates this model: subposes are represented by fully opaque image patches, and the edges between them by coloured lines.

The formal objective of the first stage of the pipeline is to take a pair of frames $(\mathbf{I}_1, \mathbf{I}_2)$ and output, for each subpose s, a subpose location $\boldsymbol{l}_s \in \mathbb{R}^2$ within the frame pair and a biposelet type $t_s \in \{1, \ldots, K\}$. Specifically, the first stage must find some $\mathbf{L} = \begin{bmatrix}\boldsymbol{l}_1 \, \boldsymbol{l}_2 \cdots \boldsymbol{l}_{|\mathcal{S}|}\end{bmatrix}$ and $\boldsymbol{t} = \begin{bmatrix}t_1 \, t_2 \cdots t_{|\mathcal{S}|}\end{bmatrix}$ which minimises the cost

$$C(\mathbf{L}, \boldsymbol{t} \mid \mathbf{D}_{12}) = w_0 + \sum_{s \in \mathcal{S}} \phi_s(\boldsymbol{l}_s, t_s; \mathbf{D}_{12}) + \sum_{(s_1, s_2) \in \mathcal{E}} \psi_{s_1 s_2}(\boldsymbol{l}_{s_1}, \boldsymbol{l}_{s_2}, t_1, t_2) \ , \quad (1)$$

where w_0 is a bias, each $\phi_s(\cdot)$ is an appearance term, each ψ_{s_1,s_2} is a pairwise cost, and $\mathbf{D}_{12}$ is the concatenation of the frames $\mathbf{I}_1$ and $\mathbf{I}_2$.

Recall that each pair of neighbouring subposes has a single joint in common; the aim of the pairwise terms is to ensure that neighbouring subposes are localised so that they predict similar locations for their shared joint. If j is a joint in the subpose s, which is itself located at $\boldsymbol{l}_s$ and assigned biposelet type t, then let $\boldsymbol{r}_{s,j}(\boldsymbol{l}_s, t)$ denote the mean location of joint j across both frames spanned by the subpose. If joint j is shared between subposes s_1 and s_2, then the pairwise cost can be written out in full as

$$\psi_{s_1 s_2}(\boldsymbol{l}_{s_1}, \boldsymbol{l}_{s_2}, t_1, t_2) = \langle \boldsymbol{w}_{s_1 s_2}, \boldsymbol{\Delta}\left(\boldsymbol{r}_{s_1,j}(\boldsymbol{l}_{s_1}, t_1) - \boldsymbol{r}_{s_2,j}(\boldsymbol{l}_{s_2}, t_2)\right)\rangle \ , \tag{2}$$

where $\boldsymbol{\Delta}(\begin{bmatrix}\delta_x & \delta_y\end{bmatrix}) = \begin{bmatrix}\delta_x^2 & \delta_x & \delta_y^2 & \delta_y\end{bmatrix}$ is a deformation feature, and $\boldsymbol{w}_{s_1 s_2}$ is a learnt weight vector.

Each appearance term $\phi_s(\boldsymbol{l}_s, t_s; \mathbf{D}_{12})$ reflects the degree to which a small region around the location $\boldsymbol{l}_s$ "looks like" a subpose s with biposelet type t_s. Concretely,

$$\phi_s(\boldsymbol{l}_s, t_s; \mathbf{D}_{12}) = -w_s \log p(s, t_s \mid \mathbf{D}_{12}(\boldsymbol{l}_s)) \ , \tag{3}$$

where w_s is a learnt weight and $\mathbf{D}_{12}(\boldsymbol{l}_s)$ denotes a crop of the frame pair $(\mathbf{I}_1, \mathbf{I}_2)$ and the optical flow between them at location $\boldsymbol{l}_s$. $p(s, t \mid \mathbf{D}(\boldsymbol{l}))$ is the probability that the image patch $\mathbf{D}(\boldsymbol{l})$ contains a subpose s with biposelet t, with the special values $(s, t) = (0, 0)$ denoting a background patch with no subpose.

$p(s, t \mid \mathbf{D}(\boldsymbol{l}))$ is produced by a two-stream convolutional neural network with an architecture loosely following the 16 layer architecture of Simonyan et al. [17]. Specifically, Simonyan et al.'s original single-stream architecture has been split in two, with one stream processing a stacked pair of RGB video frames and the other processing the flow; the two streams are merged by concatenating them after the third pooling layer. The output is a probability distribution over all subposes s and biposelets t for each subpose. This is illustrated in part (a) of Figure 1.

At test time, the network can be used as an efficient sliding window detector by converting the final dense layers to convolutions, which yields a fully convolutional network [18]. Section 5 discusses training-time considerations.

3.3 Inference on frame pair model

The full cost (1) can be minimised efficiently by exploiting the subpose graph's tree structure. For each subpose s, define the minimal cost of the subtree rooted at s as

$$\begin{aligned} M(\boldsymbol{l}_s, t_s; \mathbf{D}_{12}) = \sum_{s' : \mathrm{pa}(s') = s} \min_{\boldsymbol{l}_{s'}, t_{s'}} \left(\psi_{s s'}(\boldsymbol{l}_s, \boldsymbol{l}_{s'}, t_s, t_{s'}) + M(\boldsymbol{l}_{s'}, t_{s'}; \mathbf{D}_{12})\right) \\ + \phi_s(\boldsymbol{l}_s, t_s; \mathbf{D}_{12}) \ , \end{aligned} \tag{4}$$

where $\boldsymbol{l}_s$ and t_s are the location and biposelet type, respectively, of s, and $\mathrm{pa}(s') = s$ iff subpose s is the parent of subpose s' in $\mathcal{G}$.

M can be calculated for each location and type of the root subpose by proceeding from the leaves of $\mathcal{G}$ upwards: for each leaf subpose s_l, $M(\boldsymbol{l}_{s_l}, t_{s_l}; \mathbf{D}_{12})$ is simply $\phi_{s_l}(\boldsymbol{l}_{s_l}, t_{s_l}; \mathbf{D}_{12})$. For a non-leaf subpose s, $M(\boldsymbol{l}_s, t_s; \mathbf{D}_{12})$ can be computed by first evaluating $M(\boldsymbol{l}_{s_c}, t_{s_c}; \mathbf{D}_{12})$ for each child s_c of s, then applying Felzenszwalb and Huttenlocher's distance transform technique [19] to find the $\boldsymbol{l}_{s_c}$ which minimises $\psi_{ss_c}(\boldsymbol{l}_s, \boldsymbol{l}_{s_c}, t_s, t_{s_c})$ for each t_{s_c}.

If there are N locations in the image and K possible biposelet types, then applying this procedure to a non-leaf node will take $\mathcal{O}(NK^2)$ time for each child and each $\boldsymbol{l}_s$. Repeating the procedure for all subposes in the subpose graph $\mathcal{G}$ thus takes $\mathcal{O}(|\mathcal{S}|NK^2)$ time. Intuitively, this means that introducing new subposes to $\mathcal{S}$ or scaling up the number of pixels N is "cheap", but increasing the number of biposelet types drives up computational cost rapidly.

Having evaluated (4) for each possible root location $\boldsymbol{l}_{s_R}$ and root type t_{s_R}, finding the pose configuration which minimises (1) is a simple matter of looking up

$$\operatorname{argmin}_{\boldsymbol{l}_{s_R}, t_{s_R}} M(\boldsymbol{l}_{s_R}, t_{s_R}; \mathbf{D}_{12}) \tag{5}$$

and then backtracking to recover the locations and types of all other subposes. This backtracking process can be repeated for several root locations and types to produce a set of low-cost pose configurations for each frame pair; having several such configurations is important during the sequence stitching stage of the pipeline, as explained in Section 4.

After determining a location and biposelet type for each subpose, a location can be recovered for each joint in the original pose model using the stored joint offsets associated with each assigned biposelet. In cases where two subposes share a joint, the final joint location is the average of the joint locations predicted by the biposelets associated with those two subposes.

4 Sequence stitching

Evaluating the first stage of the pipeline on an entire video and thresholding detections by score produces a small candidate set of pose pairs for each frame pair. The second stage of the pipeline attempts to pick a single pose pair from each candidate set of pose pairs, then combines poses from overlapping pairs to obtain a single pose for each frame. Chosen pose pairs should meet two criteria:

1. Individual plausibility: low-cost poses, in the sense of the cost function (1) for the frame-pair model, should be preferred over high cost ones.
2. Temporal consistency: when two selected pose pairs overlap on a frame, the poses in that shared frame should be similar.

The pair selection criteria are easy to express mathematically. For a sequence of F frames, let $\mathcal{P}_f$ denote the set of pose pairs detected between frame f and $f+1$, where $f \in \{1, \ldots, F-1\}$. Further, let $(\mathbf{p}_{f,1}, \mathbf{p}_{f,2})$ denote the specific pose pair which the sequence stitcher chooses from set $\mathcal{P}_f$, and c_f be cost (1) associated with the subpose locations and types defining that pose pair. The objective of the

stitcher is to find a sequence of pose pairs $(\mathbf{p}_{1,1}, \mathbf{p}_{1,2}), (\mathbf{p}_{2,1}, \mathbf{p}_{2,2}), \ldots, (\mathbf{p}_{f-1,1}, \mathbf{p}_{f-1,2})$ from the sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{F-1}$ which minimises

$$\sum_{f=1}^{F-1} \|\mathbf{p}_{f,2} - \mathbf{p}_{f+1,1}\|_2^2 + \lambda \sum_{f=1}^{F-1} c_f \ . \tag{6}$$

The first term encourages temporal consistency, whilst the second favours low-cost pose pairs over high-cost ones. λ is a constant which balances the two considerations.

Once an appropriate sequence of pose pairs has been chosen, a final pose may be produced for each frame f, for $1 < f < F$, by averaging $\mathbf{p}_{f-1,2}$ and $\mathbf{p}_{f,1}$. In the first and last frames, there will only be one selected pose to begin with, as there is only one pose pair associated with each of those frames. This process is illustrated in part (c) of Figure 1.

The stitching cost (6) can be efficiently minimised using a dynamic programming approach analogous to the Viterbi algorithm. If there are $|\mathcal{P}|$ pose pairs in each candidate set, then this minimisation will take $\mathcal{O}(F|\mathcal{P}|^2)$ time. Thus, it is important to produce only a small set of the best pose pairs during inference on the frame pair model. In practice, setting $|\mathcal{P}|$ between 100 and 1000 generally gave an ample selection of poses without imposing a significant computational burden.

5 Learning

The full pipeline contains a number of learnable parameters, including CNN weights, biposelet centroids, and weights for the frame-pair cost in (1); this section describes how each set of parameters is learnt.

CNN weights: The CNN is trained to predict the joint biposelet-subpose distribution for fixed-size crops of the training data using a cross-entropy classification objective. Positive (subpose-containing) crops are taken around ground truth subposes in the training set, with a small margin around the subpose to ensure that all joints are in view. Negative (background) crops are initially taken from regions of frame pairs in the training set not containing any people.

Preliminary experiments showed that using only pure background crops as negatives left the CNN unable to discriminate between subposes in the centre of the receptive field and subposes at the edges. That significantly harmed localisation accuracy, as the CNN would predict certain biposelets with high confidence even when the true centre of the biposelet lay far from the centre of the receptive field. To rectify this issue, the CNN training code also produces more challenging negative crops which include off-centre subposes that do not correspond well (in terms of L_2 distance) to any of the learnt biposelets.

Past work has shown that two-stream CNNs are prone to overfitting [20]; in this work, overfitting is addressed with aggressive use of dropout and dataset augmentation, including random rotations, flips, and small translations. Random

scaling was not found to improve network performance, and the use of large random translations is precluded by the need to keep an entire subpose in view for the sake of accurate biposelet prediction.

To improve convergence, the network is initialised with the weights of a VGGNet trained for ILSVRC classification.[1] The same weights are applied to both the flow and RGB streams of the network, with filters for the RGB input layer duplicated to accommodate the change from three image input channels to six.

Biposelet centroids: Whenever creating a positive training sample for the CNN, the training code also produces locations for each joint relative to the crop used to make the sample. After all positive samples have been created, a set of biposelets can be produced for each subpose by clustering the crop-relative coordinates using K-means. The clustering process also yields a biposelet type for each training sample, which can then be used as a target label for the CNN. Clustering on the same samples used to train the CNN ensures that the learnt biposelets reflect the range of augmentations applied to the data, and makes it less likely that some biposelets will be underrepresented (or not represented at all) in the CNN training set.

Weights for frame-pair model: The frame-pair cost (1) can be written as an inner product between a feature vector comprised of deformation and appearance terms and a weight vector composed of the bias w_0, the deformation parameters $\{w_{s_1 s_2} : (s_1, s_2) \in \mathcal{E}\}$ and the appearance parameters $\{w_s : s \in \mathcal{S}\}$. Hence, it is possible to learn the weights using a structural SVM formulation.

Concretely, the SVM is initialised with an intuitively reasonable set of weights, which are used to perform pose estimation over the entire training set. This produces a set of positive feature vectors; negative feature vectors are produced by applying the same procedure to images from the human-free portion of the INRIA Person dataset [21]. The SVM can then be trained to label the positive feature vectors with $+1$ and the negatives with -1.

Minimisation of the SVM objective is performed using an existing dual coordinate descent solver [22]; one iteration of this detect-and-optimise process was found to be sufficient to train the model. In contrast to the work of Chen et al. [9], this work does not use location or type supervision, as neither were found to improve the learnt weights.

Stitching parameters: In principle, the λ used to balance appearance and temporal consistency during pose sequence stitching can be also be learnt using a simple global optimisation method like randomised search. In practice, stitching performance was largely invariant to choices of λ within several orders of magnitude of the (presumed) global optimum. As a result, the values of λ used in Section 6 were kept at a constant, intuitively reasonable value across all datasets.

[1] `https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3`

	Upper arms			Lower arms		
PCP threshold	0.3	0.5	0.8	0.3	0.5	0.8
Chen & Yuille [9]	32.31%	75.91%	94.21%	48.30%	73.55%	85.19%
Cherian et al. [6]	27.66%	57.14%	78.04%	30.26%	50.32%	61.44%
Combined [6] and [9]	31.63%	76.91%	95.02%	50.94%	76.29%	87.76%
Pfister et al. SpatialNet [10]	51.89%	72.50%	83.11%	36.90%	54.36%	65.37%
Biposelets	59.46%	81.88%	90.17%	31.97%	52.25%	69.03%

Table 1. PCP at various thresholds on Poses in the Wild.

6 Experiments

To evaluate the effectiveness of the model, it was applied to three continuous pose estimation datasets: Poses in the Wild, MPII Cooking Activities, and Human3.6M. The following parameters were used for all datasets:

1. Seven subposes were used by the frame pair model: one for each forearm (one left and one right), one for each elbow, one for each upper arm, and one root subpose containing both shoulders.
2. $K = 100$ biposelet types were learnt for each subpose.
3. $|\mathcal{P}| = 300$ candidate pose pairs were extracted from each frame pair.
4. $\lambda = 10^5$ was used during stitching.

Localisation error is measured using variants on Percentage Correct Keypoints (PCK) and Percentage Correct Parts (PCP) [23]. Except where hip and shoulder locations are available for normalisation, unnormalised PCK is used: that is, a joint is considered to be correctly localised for the purpose of PCK calculation if its predicted position in the image falls within a certain number of pixels of the ground truth location. For PCP, a limb of length l is considered to be correctly localised at a threshold t if both predicted endpoints fall within a distance tl of their true locations; in the case of $t = 0.5$, this is simply ordinary strict PCP.

Code for all experiments is available online.[2] Note that the experiments were performed on a machine with 128GB of main memory and 12GB of video memory per GPU; a similarly capable machine will be required to train and evaluate the convolutional networks used here.

6.1 Datasets

FLIC and PIW: Poses in the Wild (PIW) [6] is a video pose estimation dataset with frames drawn from Hollywood movies. Scene clutter, camera motion, sub-

[2] `https://github.com/qxcv/joint-regressor`

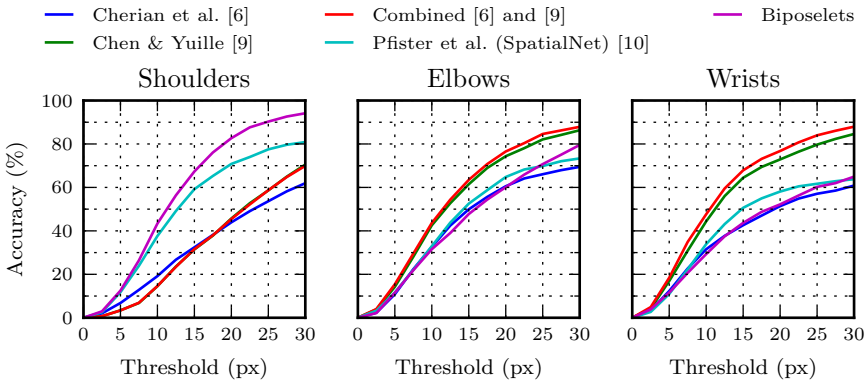

Fig. 3. Unnormalised PCK on Poses in the Wild. The curves for Chen & Yuille and the combined method occlude one another in the shoulder plot.

ject occlusion and rapid subject motion all make the dataset a challenging benchmark.

Unfortunately, at under a thousand frames, Poses in the Wild is not large enough to both test and evaluate on. Thus, the model was trained on the Frames Labelled in Cinema (FLIC) dataset [24] and then evaluated on PIW. As adjacent, labelled frames are required to train the frame-pair model, training was performed on a subset of FLIC-full consisting of around 8000 reliably annotated pairs. The remaining pairs had a mixture of incorrect annotations and excessive occlusion which made them unsuitable for training the appearance model.

All baselines used for comparison were produced using publicly released evaluation code and models trained on FLIC. Note that the "combined" baseline works by applying Chen & Yuille's [9] detector to each frame of a video sequence, then combining the produced candidate pose sets for each frame into a pose sequence using the recombination method of Cherian et al. [6]. It should also be noted that the Pfister et al. baseline is for a non-temporally-aware model, which they dub "SpatialNet", rather than the flow-augmented model for which Pfister et al. obtained the best results; code for the latter model was not publicly available at the time that this paper was written.

PCK at image scale over all of Poses in the Wild is shown in Figure 3, while PCP at various thresholds is shown in Table 1.

MPII Cooking Activities: MPII Cooking Activities [16] is a kitchen-themed action recognition dataset which includes two pose estimation benchmarks. The Cooking Activities data represents a near-ideal case for video pose estimation: all videos are recorded from the same static camera in the same kitchen, with minimal occlusion and only one subject per video sequence.

Training is performed on the "continuous pose estimation" dataset released with MPII Cooking Activities, while the "pose challenge" dataset is used for evaluation. It should be noted that the *training* set of the pose challenge is being used for evaluation because the test set does not have continuous frames; despite

	Upper arms			Lower arms		
PCP threshold	0.3	0.5	0.8	0.3	0.5	0.8
Chen & Yuille [9]	79.44%	95.07%	98.39%	80.03%	96.00%	97.90%
Cherian et al. [6]	67.97%	80.91%	88.62%	55.71%	75.93%	83.45%
Combined [6] and [9]	79.79%	95.07%	98.58%	81.54%	96.97%	99.07%
Biposelets	76.59%	87.44%	94.72%	69.21%	85.43%	94.28%

Table 2. PCP at various thresholds on MPII Cooking Activities.

the confusing nomenclature, the training set of the "pose challenge" (which is used for evaluation) is not the same as the "continuous pose estimation" dataset (which is used for training).

As with the comparison on PIW, all baselines used here were produced using publicly released models trained for the FLIC dataset. Significant difficulty was encountered in getting Pfister et al.'s publicly released SpatialNet model to produce competitive results on MPII Cooking Activities, so the SpatialNet baseline was omitted from the Cooking Activities results.

PCK curves for MPII Cooking Activities are given in Figure 4, and PCP at various thresholds is given in Table 2.

Human3.6M: Human3.6M [25,26] is a pose estimation and action recognition dataset which includes full depth and 3D pose data recorded with a motion capture system, although this evaluation uses only the RGB video and 2D pose portions of the dataset. As a motion capture dataset, Human3.6M is recorded in a controlled environment with a uniform background and no camera motion. However, in other respects, it is significantly more challenging than PIW or MPII Cooking Activities: not only do its actors exhibit a wider range of motion, but all

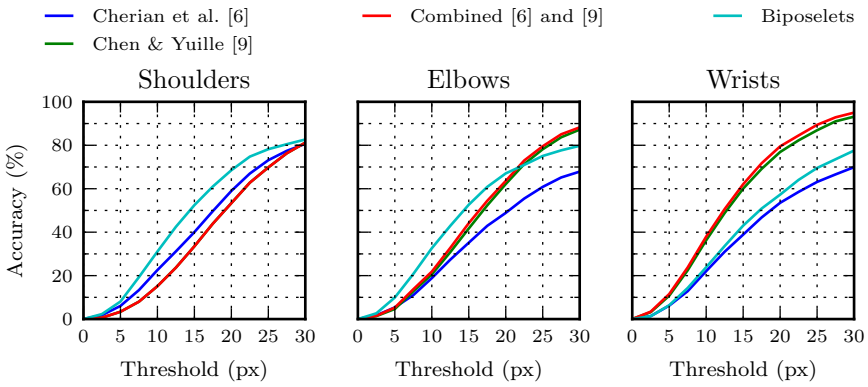

Fig. 4. Unnormalised PCK on MPII Cooking Activities.

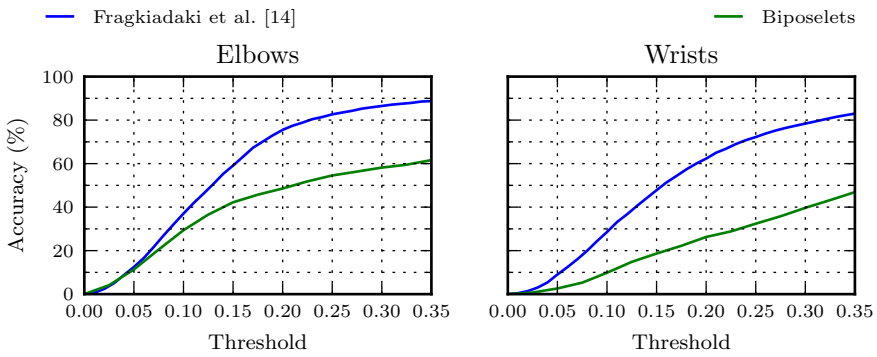

Fig. 5. PCK for subject five on the Human3.6M dataset. Thresholds are fractions of the distance between the subject's left hip and right shoulder.

motion is recorded from four cameras spaced evenly around the scene, meaning that a significant portion of poses are non-frontal.

The primary motivation behind evaluating on Human3.6M is to compare to the recent results reported by Fragkiadaki et al. [14] for a sophisticated recurrent neural network architecture. The training and evaluation protocol used for the biposelet model is similar to that employed by Fragkiadaki et al. with subject five used for evaluation and all other subjects used for training. Due to time constraints, it was not possible to evaluate the biposelet model on all frames associated with subject five. Instead, the statistics given here are for a randomly selected set of half the 120 available scenes, each of which has been further trimmed to 5% of its original length. PCKs for both the biposelet model and the recurrent network approach are given in Figure 5.

6.2 Discussion

The results for Cooking Activities and PIW show that the biposelet model improves significantly on past work in the localisation of shoulders, and is competitive on elbow localisation; this is reflected in both PCKs and in the difference between upper arm and forearm PCPs. The use of subposes is likely responsible for the high performance on shoulders: As mentioned in Section 1, identifying an entire subpose can be easier than identifying a single joint—which is what all the baseline methods do—because the CNN is able to make use of more surrounding context. This is particularly useful for shoulders, since the upper part of the torso is almost always in view whenever both shoulders are, and the torso's size and consistent appearance make it easy to identify [27]. As is explained later in this section, arms may not benefit as much from the use of subposes because of their higher level of articulation.

Despite the model's excellent overall performance in localising shoulders, qualitative analysis revealed that shoulders were very poorly localised in a small subset of the Cooking Activities frames. Frames (h) and (i) in Figure 6 illustrate

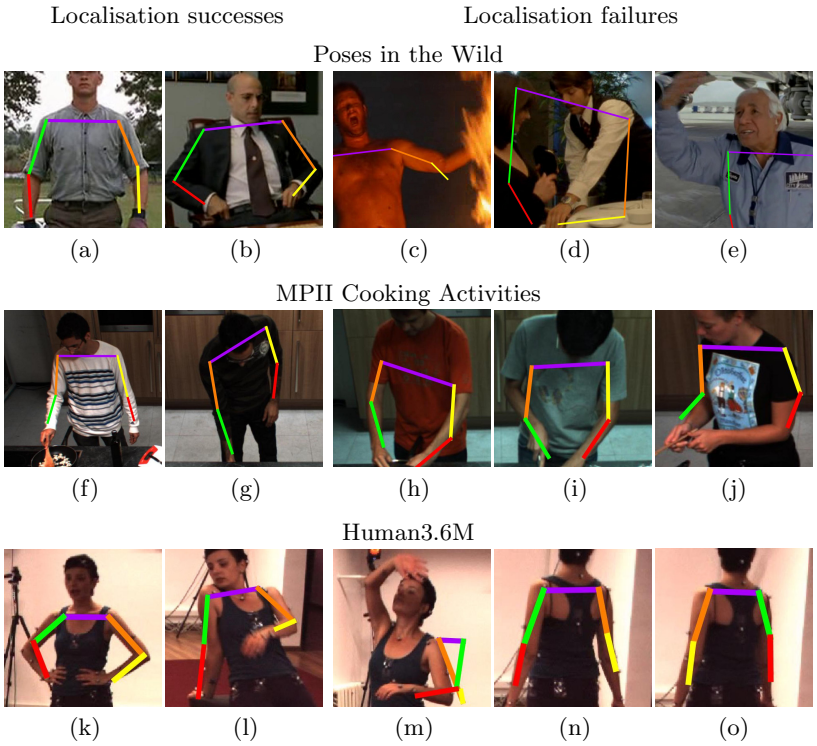

Fig. 6. Characteristic successes (first two columns) and failures (last three columns).

this problem. The fact that the same issue does not crop up in the more challenging PIW dataset suggests that the training protocol for Cooking Activities may be flawed. A possible culprit is the lack of diversity in the subset of Cooking Activities used for training—Cooking Activities uses only a handful of actors, so the CNN may be overfitting to aspects of their appearance when learning to classify shoulders. This problem may be ameliorated by training on FLIC—as was done for the PIW evaluation—rather than on a subset of Cooking Activities.

A major weakness of the model across all datasets was in handling high levels of joint articulation. Nowhere was this more evident than in the wrists—examples (e), (j) and (m) in Figure 6 are representative of the many wrist localisation failures observed during evaluation. Such failures may be due to the limited number of joint configurations which can be expressed using a small set of biposelets; the baseline models do not suffer from this problem to the same degree because they localise individual joints directly, rather than localising entire subposes and then attempting to extract specific joint locations using a discrete set of subpose configurations.

Biposelets must be able to express both joint location and motion, so it may be that biposelet sets which are capable of expressing a wide range of motion

are less capable of expressing fine differences in location. That would partially explain why forearms are localised so poorly relative to upper arms: wrists and elbows undergo a higher degree of motion than shoulders, so it is expected that their associated biposelets should be able to express more extreme motion than the biposelets learnt for shoulders. Experimenting with higher numbers of biposelets and dividing the forearms into more subposes could address this deficiency of the model.

Another possibility is that optical flow is not being used as effectively as it could be. Wrists can move much faster than elbows or shoulders, so the magnitude of optical flow in an image patch should be a strong cue for the presence of wrists. This is clearly illustrated by the "combined" baseline in Figure 3 and Figure 4: combining Chen & Yuille's approach with Cherian et al.'s (largely flow-based) recombination heuristics yields the greatest performance improvement for wrists, and no performance improvement for shoulders. Although the biposelet model uses raw optical flow at the CNN input layer, Jain et al. [5] suggest that use of raw flow can lead CNNs to overfit, and that supplying only the magnitude of flow may be more effective.

Much like high level of joint articulation, occlusion was a significant challenge. Frames (c) and (d) in Figure 6 show occlusion-related failures: in (c), the presence of a flame occluding the subject's left wrist has led their entire forearm to be improperly localised, whilst in (d), two subjects occlude one another, and the detector attempts to fit the same pose to both of them.

Poor performance on occlusions reflects a deficiency of the model: apart from the pairwise deformation features in the cost function used for frame pair inference, the model does not have any way of reasoning about the position and type of an occluded subpose based on the appearance of its visible neighbours. Improving performance on subposes which are entirely occluded will likely require the use introduction of something like Chen & Yuille's image-dependent pairwise relations [9], which enable reasoning about the position of a joint (or subpose, in this case) based on image evidence at neighbouring joints (or subposes). In contrast, deformation features depend only on the relative positions of subposes, and not on the appearances of the subposes themselves.

Localisation performance on Human3.6M was uniformly poor relative to the baseline. Visual inspection revealed that a majority of errors were due to confusion between the left and right arms. For example, poses (n) and (o) in Figure 6 show a situation in which the estimator reversed its labelling of the left and right arms within the space of two frames, even though the subject remained facing in the same direction. Moreover, left–right confusion manifested itself in instances like (m), where both the left and right arm were predicted to lie in the same place.

Left–right confusion is a common problem in pose estimation; indeed, Fragkiadaki et al. [14] note that their recurrent model outperforms a per-frame baseline precisely because it is so effective at resolving left–right confusion. As future work, it may be useful to augment the graphical model in Section 3 with a la-

tent variable indicating whether a person is forwards-facing or backwards-facing, much as Sapp & Taskar do to eliminate the left–right confusion [24].

7 Conclusion

Biposelets have proved to be effective in boosting pose estimation performance on shoulders and, to a lesser extent, elbows. A biposelet-based approach is able to make effective use of the distinctive visual context of those joints by localising entire subposes instead of identifying a single joint at a time. Predicting poses in pairs also leads to an elegant formulation for stitching pose predictions together in a video setting, and forces the model to make use of the visual context present in two video frames at a time. The price of these practical and theoretical improvements has been a drop in localisation accuracy for faster-moving joints; this likely stems from the limited ability of biposelets to represent both the motion and the position of those joints. Nevertheless, the already-competitive performance of a biposelet approach and the opportunities for improvement enumerated in Section 6.2 make biposelets a promising area for future research.

Acknowledgements

I would like to thank various authors whose code I have used to produce baseline comparisons [10,9,6]. I would also like to thank Anoop Cherian for informing many of the ideas presented here and giving extensive feedback on drafts of this paper.

References

1. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D pose estimation and tracking by detection. In: CVPR. (2010)
2. Zhou, F., De la Torre, F.: Spatio-temporal matching for human pose estimation in video. TPAMI (2016)
3. Chéron, G., Laptev, I., Schmid, C.: P-CNN: pose-based CNN features for action recognition. In: ICCV. (2015)
4. Yao, A., Gall, J., Fanelli, G., Van Gool, L.J.: Does human action recognition benefit from pose estimation? In: BMVC. (2011)
5. Jain, A., Tompson, J., LeCun, Y., Bregler, C.: MoDeep: A deep learning framework using motion features for human pose estimation. In: ACCV. (2014)
6. Cherian, A., Mairal, J., Alahari, K., Schmid, C.: Mixing body-part sequences for human pose estimation. In: CVPR. (2014)
7. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR. (2011)
8. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS. (2014)
9. Chen, X., Yuille, A.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS. (2014)

10. Pfister, T., Charles, J., Zisserman, A.: Flowing convnets for human pose estimation in videos. In: ICCV. (2015)
11. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. arXiv:1602.00134 (2016)
12. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR. (2008)
13. Ramanan, D., Forsyth, D.A., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: CVPR. (2005)
14. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: ICCV. (2015)
15. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3D human pose annotations. In: ICCV. (2009)
16. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR. (2012)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
18. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229 (2013)
19. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Theory of Computing **8** (2012) 415–428
20. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv:1507.02159 (2015)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
22. Ramanan, D.: Dual coordinate solvers for large-scale structural SVMs. arXiv:1312.1743 (2013)
23. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. TPAMI (2013)
24. Sapp, B., Taskar, B.: MODEC: Multimodal decomposable models for human pose estimation. In: CVPR. (2013)
25. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments. TPAMI (2014)
26. Ionescu, C., Li, F., Sminchisescu, C.: Latent structured models for human pose estimation. In: ICCV. (2011)
27. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Volume 2., IEEE (2004) II–326

The Australian National University
School of Computer Science

THE AUSTRALIAN NATIONAL UNIVERSITY

Independent Study Contract

Note: Enrolment is subject to approval by the projects coordinator.

Section A (Student & Supervisor(s))

UniID: U5568237

Surname: TOYER First names: SAMUEL DAVID

Project supervisor (may be external): ANOOP CHERIAN

Course supervisor (a SoCS academic): WEIFA LIANG

Course code, title and unit: COMP3710, TOPICS IN COMPUTER SCIENCE

Semester: ☑ S1 ☐ S2 Year: 2016

Project title:

Efficient and Effective Pose Estimation using Deep Learning

Learning Objectives:

To gain experience
(a) Working with state-of-the-art computer vision techniques
(b) Producing papers suitable for submission to leading conferences or journals.

Project Description:

Human pose estimation is the task of automatically annotating the locations of a person's limbs within an image or video. Occlusions, rapid motion, and the sheer diversity of poses which humans can exhibit all make this a challenging task, and it has received significant attention from the computer vision research community for more than a decade. This project aims to explore novel approaches to human pose estimation using state-of-the-art vision techniques, with a focus on applications of deep learning to the problem.

Assessment (as per course's project rules web page, with the differences noted below):

Assessed project components:	% of mark	Due date:	Evaluated by:
Report: name style: Research paper (e.g. research report, software description, ...)	60% (60%)	27 May	
Artefact: name kind: Code for reproducing results in paper (e.g. software, user interface, robot, ...)	30% (30%)	27 May	
Presentation: Presentation to research groups	10% (10%)	17 May	

Meeting dates (if known): Research group updates weekly on Wednesdays; meetings with supervisor as needed.

Student declaration: I agree to fulfil the above defined contract:

Signature ..... Date 03/03/2016

Section B (Supervisor)

I am willing to supervise and support this project. I have checked the student's academic record and believe this student can complete the project:

Signature ..... Date 03/03/2016

Required department resources:

Section C (Course coordinator approval)

Signature ..... Date 3-3-16

Section D (Projects coordinator approval)

Signature ..... Date