Imperial College London

Interim Report

Imperial College London

Department of Electrical & Electronic Engineering

PoseReAct: Cyclic Refinement of Depth-Based 3D Hand Pose Estimation and Egocentric Action Recognition

Author:
Haaris Osman Mehmood

Supervisor:
Dr. T-K Kim

Second Marker:
Professor Jeremy Pitt

Monday 28th January, 2019

Contents

Chapter 1

Introduction

1.1 Overview

Human action recognition has long been a core challenge in the computer vision community with many practical applications arising as a direct result of the contributions made in this field. Traditional models have focused on hand-crafted features with many robust techniques proposed and consequently adopted by the industry. With the rapid rise of deep-learning based prediction models, action recognition has too conquered new frontiers with interests now shifting towards predictions for future actions.

In parallel, with the recent developments of wearable video recording devices such as Go Pro [1] and SnapCam [2], it is evident that the way we traditionally have been using the camera lens is about to change. 'Vlogging' [3] is another modern day phenomena which can be expected to be the first practical use cases of such devices. Wearable cameras record videos in whats known as 'egocentric' or 'first-person view'. The introduction of such devices in the market has subsequently led to several researchers now working towards egocentric video activity recognition and help pave the way for future applications in this domain.

Traditionally, everyday cameras have only been able to record RGB images and videos but with the recent availability of consumer-focus depth sensors (e.g. Intel RealSense 3D [4]), this too is about to change. It leads to the assumption that such device will soon become increasingly popular and in demand as they become increasingly portable (e.g. as part of HoloLens [5]). As a result, its not unrealistic to expect a demand for depth-based wearable cameras in the near future.

Performing action recognition in the first-person domain is considered much more challenging than its third-person counterpart. In first-person view, the source actor is now also the camera-person. Thus we can already expect video recordings to have a inherent 'shaky-camera' effect. Furthermore, most of the actor joints visible in third-person view is usually occluded in first-person view and only a limited part of his/her body is visible.

Previous work [6, 7] has shown that hands, being the most likely visible part of an actor in first-person view are also a source of strong discriminative signal for action recognition. This project will revolve around the use of hands as the main source of information to predict the action that a first-person actor is currently performing.

1.2 Objectives

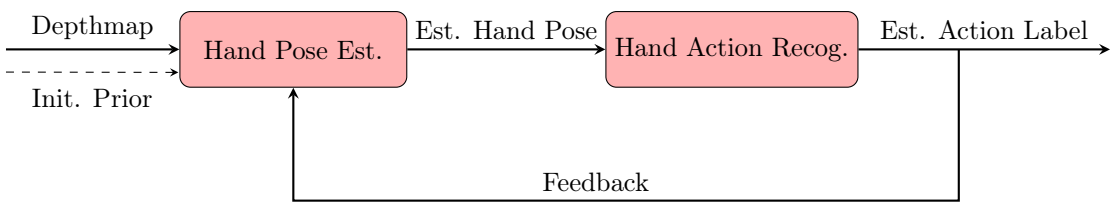


Figure 1.1: An block-diagram of the proposed framework.

The main aim of this project is to create a model that is able to produce two outputs of different modalities given an input of a third modality. It particular, we focus towards regressing hand poses and predicting (classifying) first-person actions with the use of a sequence of depth frames originating from a video clip demonstrating a particular action.

The model should architected to enable the use of the first output modality (hand pose joints) as an additional prior (or the only prior) to predict the second output modality (action class label).

Furthermore, we wish to explore the feasibility of a generalised feedback-based framework (a cyclic model), much similar to the structure of a vanilla recurrent neural network (generic single-cell RNN unit) [8], to iteratively reduce the error on estimation of either outputs by using the estimate of one output to improve the other and vice versa.

Lastly, understanding that imposing priors for the purpose of improving deterministic outputs is a harder problem and may lead to a lesser gain in performance, we wish to exploit generative (stochastic in particular) models (variational inference [9], adversarial modelling [10] and test-time dropouts [11] to name a few) that may be adopted to include additional priors to produce a prior conditioned posterior e.g. predicting action label using depth-map sequence as an input given a key-point sequence as a prior.

1.3 Formulation

1.3.1 Notation

$(D_1 \times D_2) = 128 \times 128$	Depth Frame Resolution
$J = 21$	Recorded/Estimated Hand Joints
$C = 45$	Action Classes
F	Frames Per Action Sequence
$\mathcal{U} \subseteq \mathbb{R}^{(D_1 \times D_2)}$	Depth Subspace
$\mathcal{V} \subseteq \mathbb{R}^{(3J)}$	Hand Skeleton Subspace
$\mathcal{W} \subseteq \mathbb{R}^{C}$	Action Class Probability Subspace
$\boldsymbol{U} \in \mathcal{U}$	Depth Frame
$\boldsymbol{v} \in \mathcal{V}$	Hand Skeleton/Pose Vector
$\boldsymbol{w} \in \mathcal{W}$	Action Class Probability Vector
$\mathcal{U} \subseteq \mathbb{R}^{30}$	Depth Latent Subspace
$\mathcal{V} \subseteq \mathbb{R}^{30}$	Hand Skeleton Latent Subspace

1.3.2 Model

$$\mathcal{U} \xrightarrow{f} \mathcal{V} \underset{h}{\overset{g}{\rightleftarrows}} \mathcal{W} \tag{1.1}$$

In essence, the model can be described as training and optimising three universal function approximators (such as Neural Networks) either in conjunction or separately. The function $f(\cdot)$ can be any generic hand-pose estimator or generator model to transform depth-maps to hand-poses. The function $g(\cdot)$ will be used to map a sequence of hand-poses corresponding to a given task (and thus constant action class C). This can be performed using any generic action recognisers including baseline models such as a single-cell LSTM. Lastly, the function $h(\cdot)$ will perform the required transformation to improve the estimation of depth-maps using action class information. Although not shown for brevity, it is implied that $h(\cdot)$ will make use of both the ground-truth depth-map information and the newly predicted action class information to improve hand-pose estimation.

1.3.3 Dataset

As a whole, the training requirements of our proposed model is that it requires ground truth annotations of iid (independent and identically distributed) triplets rather than iid pairs:

$$
\begin{aligned}
\mathcal{D} &= \{(\mathbf{U}_1, \boldsymbol{V}_1, \boldsymbol{w}_1), ..., (\mathbf{U}_N, \boldsymbol{V}_N, \boldsymbol{w}_N)\} \\
\mathbf{U}_i &= [\boldsymbol{U}_1, ..., \boldsymbol{U}_F]_i \in \mathbb{R}^{F \times D_1 \times D_2} \\
\boldsymbol{V}_i &= [\boldsymbol{v}_1^T, ..., \boldsymbol{v}_F^T]_i \in \mathbb{R}^{F \times 3J} \\
\boldsymbol{w}_i &= [p(a_1), ..., p(a_C)] \in \mathbb{R}^C \\
&\sum_{j=1}^{C} p(a_j) = 1 \\
&0 \leq i \leq N \\
&N = |\mathcal{D}|
\end{aligned}
\tag{1.2}
$$

However during inference only one annotation, $\mathbf{U}_i$ (depth-information), is present. This makes the learning problem and the overall framework very close to the 'Learning Using Privileged Information' (LUPI) as described in [12, 13]. Given our objectives differ from LUPI, we do not explore the LUPI framework in greater detail but instead just use it as a basis for the definition of our problem.

1.4 Motivational Insights

Some motivations for this project has been described in 1.1. We now look at other aspects of the idea of using a setup as described in 1.2 for the purpose of first-person action recognition.

The two-part models described in 1.1 is inspired from previous works that have shown that human skeleton (also known as joints or key-points or pose) information is a strong source of 'side-information' for the purpose of action-recognition from depth images [14, 15]. One reason behind this is that 3D pose annotations are view-point invariant whereas depth-maps suffer severely with occlusions due to extreme viewpoints (e.g. looking at the hand from the side). To our advantage, recent advancements in depth-pose estimators have made them favourable to infer 3D information from depth-maps and use this for action-recognition.

Our reasoning for formulating the model as a two-part solution rather than a single framework is because we wish to output two different modalities much similar to 'multi-task learning'. On top of that, we wish to explore general frameworks that can be used in conjunction with existing popular hand-action recognisers and hand-pose estimators making it useful for the existing community working in both such areas.

Looking at Table 4 of a recent evaluation for RGB, depth and pose based action recognition [16] we can say that most pose-based estimators tend to perform superior than other modalities. For practical reasons, it is very hard to get accurate key point information in real-time as it requires accurate magnetic sensors placed over the body so for everyday tasks this cannot be achieved in any way.

To overcome this impracticality, researchers have increasingly worked towards inferring key-point information from easy to record modalities such as RGB [17], depth [18] or both [19]. Thus as described earlier, we can say that the mapping from $\mathcal{U}$ to $\mathcal{W}$ is a harder problem to solve than the mapping from $\mathcal{U}$ to $\mathcal{V}$.

$\mathcal{U}$ is simple the space spanned by depth-maps, analogous to RGB. Popular image recognition and video action recognition models make use of complex CNN feature extractors to map these RGB images to a lower dimension subspace which is discriminative enough (unlike pixels) for any kind of classification. This is what the last few layers of a CNN based classifier usually represents [20].

Given a model as described in 1.1 we can imagine $\mathcal{V}$ to be very similar to the 'extracted' features of CNN classifiers (this this actually the case for our baseline hand-pose estimator [18]). Thus the objective of action recognition can be split into two parts: feature extraction and then classification using the extracted features. This first part can be approximated using the mapping f in 1.1.

Now our motivational assumption is that if we have some ground truth training data that is from the same sub-space as $\mathcal{V}$, we can make use of such information as a strong source of 'side-information' during training to improve training of f (as opposed to just using samples from $\mathcal{U}$ and $\mathcal{W}$) and to some extent g. Note how both g and h would likely involve LSTM layers and thus the back-propagation loss could have vanishing gradients when it reaches the weights of f

(even though LSTMs aim to fix this problem, it still exists to a certain extent for longer temporal dependency).

Thus the use of such a model should intuitively (a) provide better results for predicting $\mathcal{W}$ using $\mathcal{U}$ and $\mathcal{V}$ than just using $\mathcal{U}$ and (b) provide better results for predicting $\mathcal{V}$ using $\mathcal{U}$ and $\mathcal{W}$. However the second improvements might not be that noticeable considering the mapping between $\mathcal{U}$ and $\mathcal{V}$ is easily realisable and there exists many state-of-art models to learn that. Furthermore there are many valid and possible $v \in \mathcal{V}$ that come under the same action class $w \in \mathcal{W}$ whereas excluding viewpoint occlusions and transformation approximation, $\mathcal{U}$ to $\mathcal{V}$ mapping is almost 1-1 and is assumed as such in some papers [21].

1.5 Safety, Legal and Ethical Issues

This project does not involve acquiring new data from human demonstrators. We will mainly work on a publicly available data-set as described in [16]. The relevant subjects in this data-set have all already agreed for the use of their data-set for research purposes and its availability in the public domain. Thus in terms of ethical concerns there are no apparent issues that could be faced during the development of this project.

All source-code provided as a deliverable and derived from other works will be based on open-source software only and as such no royalties or licences need to be paid to any party for any use of software during the course of the project. As far as we are aware this project does not infringe on any intellectual property as all third-party derivations of methods, knowledge and components that form part of the final deliverable are from prior publications and software releases that openly exist in the public domain. Where possible we will strive to source original authors in our final deliverable's repository from which methods are used either wholly or partly.

Safety-wise, this project makes no use of any direct hardware components apart from the use of standard computing devices and peripherals either locally or on the cloud. Thus, there is no underlying concern on human or animal safety during the course of this project.

Chapter 2

Background

2.1 Hand Pose Estimation

2.1.1 Overview

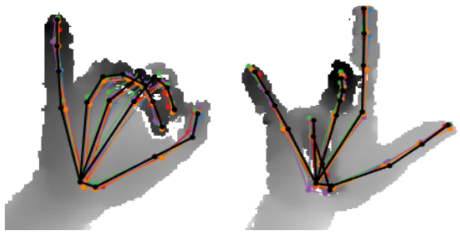


Figure 2.1: An overlay of multiple 3D hand pose estimators on 2D depthmap. Image Source: [22]

Hands are an integral part of the human body and play an important role for many primitive and modern age tasks such as grasping, eating, body language, sign language. The expressivity and fine control of our hands make it a popular candidate for modern day Human-Computer Interaction (HCI) (including keyboards, mice, and virtual intractions).

In recent years a plethora of applications have arisen that make use of 3D human pose as a key source of input to perform an action or process information in the digital domain. Pose-estimation is a key component for many consumer facing devices today with particular emphasis on the gaming industry. Furthermore, the rise of hands-free computing devices such as AR and VR headsets have further strengthen the application and need for accurate 3D hand-pose estimation for interacting with the virtual world.

In the most general sense hand pose estimation involves predicting locations of hand-joints in 3D coordinates which can thus be represented in matrix or vector form. The matrix representation describes each 3D co-ordinate (x_j, y_j, z_j) as a row vector $[x_j, y_j, z_j]^T$ and given J joints a $\mathbb{R}^{J\times 3}$ matrix is formed for each sample

The vector form flattens this matrix for easier computation and representation and due to the fact that unlike a 2D matrix of pixels representing an image, the 3D hand joints inherently encode spatial information within themselves i.e. each component of such matrix directly describes the position of a point along a specific plan in the euclidean space. We follow the notation described in 1.3.1:

$$\boldsymbol{v} = [x_1, y_1, z_1, x_2, ..., x_J, y_J, z_J]^T \tag{2.1}$$

The advent of reliable consumer devices for capturing depth information [4, 5] since the past decade has led to the rapid increase in the availability of practical and varied data-sets which in turn caused an increased interest in proposing a wide variety of approaches for hand-pose estimation [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]

A comprehensive recent study of the various methods involved in hand-pose estimation can be found in [22]. Upto-date information on hand pose estimation can be found in [35].

2.2 Approaches

Hierarchical Models

These model hand pose estimation as a multitask learning problem and pass the same input image through multiple extractors and then finally concatenate results [36, 37].

Structured Methods

A very popular approach of the past and still relevant today [23, 24, 38, 29, 18], these approaches incorporate some form of a hand model which acts as a prior. Seminal work introducing the use of kinematic model of the hand has been very popular and many variants exist since the original work [39]. We describe work in this field as the general pictorial structured model below.

Multi-Stage Refinement

Many of these methods have similar end goals as this project. These involve predicting one or more components in stages which can be later used as refinement. For e.g. [36] first predicts an initial hand-pose and then uses this estimation to extract various regions of feature maps from a region ensemble network (REN). A hierarchical approach is then used to incorporate the results from REN to improve hand pose estimation.

Generative vs Discriminative Models

Discriminative methods use data to learn to predict either heat-maps defining probabilities for each joint's existence at each pixel or regress 3D coordinates directly. They do not require any hand-crafted modelling and are suitable for real time applications. In the past, random forests based methods came out as top performers due to their highly parallel structure and robust generalisation [24, 25, 26]. However, with the rapid popularity of deep learning hand pose estimation has shifted towards CNN-based feature extractors [29].

In the past, generative methods have typically involved the use of a hand model to optimise initial pose predictions such that the likelihood of a predicted pose matching the true hand distribution is maximised [36] i.e. the estimated hand-pose is kinematically plausible.

Similarly, several other methods exists than can be considered as a combination of both discriminative and generative models [38, 27, 22]. In-fact any form of optimisation after first extracting discriminative features using a pre-trained or end-to-end model can be considered a hybrid approach. Since we aim to use either a hand-pose generator or estimator for the first stage of our model and then later refine it, our model can be considered at-least a hybrid approach if not a fully generative approach.

Newer deep generative models such as VAEs and GANs can be used to provide an output sampled from a certain conditional distribution, in this case, there is no required need for per-sample model-based optimnisation and any form of output that a traditional discriminator is able to provide (specially probability heat-maps or 3D co-ordinates) can be provided using deep generative models. The outputs of such generative models have an underlying stochasticity and this implies that multiple closely related output samples can be generated using such models that correspond to the same input. This is interesting to us since we can use this technique as a basis to explore the possibility of using multi-instance learning frameworks to improve baseline performance on hand-action recognisers (HAR).

Additionally, these models allow, to some extent, sampling from latent space which can be regarded as data augmentation in the form of unsupervised training for the purpose of improving hand-pose estimates. It is often the case that many generative models actually incorporate a discriminative component in their models such as decoders in VAEs and special decoder-like extensions added to GANs for the purpose of hand pose estimation given latent codes. These discriminators benefit quite a lot from unsupervised training based on latent space sampling [21].

Furthermore, because a large emphasis on our project is the use of action class probability priors to improve had pose estimation and this can be seen as per sample test time optimisation, there is a strong possibility that we would require the need for making use of a hand-model as a prior in our framework so as to ensure that the final generated poses do not become unrealistic due to a wrong feedback signal. Such a framework can be incorporated as shown in [22]. We explore some background on pictorial structured (PS) models below.

3D Volumetric or Point-Cloud Based Representation

Although 2D-CNN based approaches have been the norm a few years ago, very recent advances have shown the effective use of 3D-CNNs for the purpose of predicting hand-joints. Intuitively it seems natural that if we project a depth-map to 3D, a network can better learn the 3D representation of a hand and as a consequence produce better predictions. Two main methods for 3D representations include voxels [32] or sparse point-cloud representations [40]. Voxels (small uniform cubes spanning 3D space) can be seen as a natural extension to heat-maps and instead of per-pixel likelihood, a per-voxel likelihood is computed. A further extension to voxels is that their sizes can be adjusted to that would correspond to the level of precision required in predictions. This is akin to changing the resolution of 2D images. The benefit of using voxels is that unlike heatmap predictions which require a secondary step before predicting final 3D co-ordinates, voxels can be directly 'warped' to 3D euclidean space. On the other hand, point-clouds are simple projections of each pixel in a depth-map to a discretized point in the 3D space to produce a sparse representation of the hand in 3D space. Unlike voxels, point-cloud estimators directly regress 3D joints and instead of 3D-CNNs make use of specially adapted 2D-CNNs for the purpose of feature extraction from sparse 3D point clouds [41]. This allow point cloud-based models to achieve the best of both worlds: the lower training and inference complexity of 2D-CNNs and at the same time the more informative representation in 3D space.

Although 3D-CNNs give a better performance overall and capture 3D spatial information in a better way than 2D CNNs, the time and space complexity now scales cubically with volume as compared to quadratically for 2D estimation. Furthermore, since our focus is to develop on a general framework that should be broadly adaptable to different approaches in recent works, we do not seek to perform our hypothesis testing on the current best performer but rather will focus on using a strong baseline that has several desirable properties such as easy implementation, faster training and wide adoption for baseline comparison across many works. One such model that fits this criteria is Deep-Prior++ [18]. We have already ported many components of this model to a more relevant choice of framework and are soon expected to complete this port.

Heat Map vs Direct Regression

As described earlier both discriminative and newer generative models can have their results regressed to 2D or 3D co-ordinates directly and a MSE is calculated or, a heat-map spanning the entire input image resolution can be produced as a form of a pixel-wise binary classification pertaining to the probability of existence of a particular joint at a particular pixel location. The loss function for this case would be a pixel-wise sigmoid cross-entropy loss function with the input image pixels containing 1s for a particular sub-part's region in the original image and all other pixels are set to 0. Generative models typically employ heat map based outputs since its easier to impose a prior in such models. However, as described previously newer generative models can also employ regression based estimation.

2.2.1 Challenges

One of the challenges in hand pose estimation is the 'self-similarity' of the hand [18]. This implies that parts of the hand in particular the fingers are very similar to each other and thus using them as a discriminate feature is a non-trivial task. Other problems arise from occlusion of hands [42] which is even more prominent when we try to record humans interacting with everyday objects. Even without the interaction, parts of the hand can get 'self-occluded' e.g. when we close our fist.

Overall, due to the tremendous dexterity of the hand it is impossible to account for every possible pose that humans could possibly make in their everyday lives and thus there exists a demand for hand pose estimators that generalise well to unseen data, can distinguish most of the hand joints separately and are robust to occlusions.

2.2.2 DeepPrior++: A Baseline

The biggest contribution of the original Deep-Prior model [29] is its use of a separately trained 'prior' that can be transparently integrated to the main regressor model to improve hand-pose prediction. DeepPrior++ [18] implements a ResNet-50 [43] based feature extractor for the purpose

of extracting discriminative features to estimate 3D hand poses (joint locations in $((x, y, z))$) from depth maps. It also includes other updates to the original framework including further data augmentation and the use of a Refine-Net architecture.

Since hand-pose estimation is a regression task and the ResNet architecture was originally designed for classification, the final global max-pooling layer for Deep-Prior++ is removed and instead a couple of more FC layers are added to the end of the network. Furthermore, the final layer has no activation which is the norm for regression as opposed to using soft-max for classification.

An important aspect of the Deep-Prior frame-word is the use of a bottleneck layer which forces a lower-dimensional (and thus a higher level) representation of hand poses. The motivation of such bottleneck stems from previous study [44] where it has been shown that due to the severe physical limitations of hand joints, hand articulation (movement) can be safely modelled in a lower dimensional space.

The Prior

The Deep-Prior++ model makes use of a two-part model where the first part (the 'prior') can be modelled as an autoencoder which encodes J 3D joints of the hand to a lower dimension, written as:

$$p_v : \mathcal{V} \subseteq \mathbb{R}^{3J} \rightarrow \bar{\mathcal{V}} \subseteq \mathbb{R}^{30}$$

In practice, this mapping can be efficiently learnt by computing a closed form solution for the PCA of all samples present in the training set, represented by a matrix of row vectors:

$$\boldsymbol{V}_{train} = [\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, ..., \boldsymbol{v}_{N_{train}}^T] \in \mathbb{R}^{N_{train} \times 3J}$$

It can be shown [45] that the loss function (MSE) of a single-layer, linear auto-encoder has a unique minimum which corresponds to the encoder's weight matrix being exactly equal to the projection matrix spanned by the required largest principal components of $\boldsymbol{V}_{train}$ (after mean removal).

Hand Detection

Practically, to use any hand-pose estimator model in a real-time scenario or with unlabelled input data, it is essential that the approximate location of the hand within a depth image must first be detected and additionally, in the case of clutter it must be accurately segmented. This facilitates the cropping of the hand to the center of an image and ensure as much as possible background is removed from the depth image.

Prior works in hand-pose estimation consider hand detection a solved problem (in reasonablly low clutter) and often make use of stardised or pre-existing solutions to perform hand detection and segmentation [24, 46, 47, 26].

Efficient libraries exist today to easily delegate most of this work and due to the relatively moderate requirements of the most popular data-sets in use today [26, 47, 48] there isn't a strong desire for powerful deep neural network based segmentation networks for hand-cropping. However in future, we may see the need for such models to tackle data-sets with harder constraints such as increased clutter and diversity [42].

A common method to ensure that the weights of a neural network are not inclined to learn to have a high magnitude, the inputs and outputs to a neural network are usually standardised or normalise. For the case of Deep-Prior, this process is done by roughly standardising values as $u \sim \mathcal{U}(-1, 1)$ and $v \sim \mathcal{U}(-1, 1)$ where $\mathcal{U}$ represents the uniform distribution.

Binary depth thresholding [49] is a technique used to binarise an image by setting all colour intensity or depth values in the required range to 1 and the rest to 0. Later, using a contour finding algorithm [50] and assuming the hand is the largest closed contour present in the image, the center-of-mass point of this contour is found.

Although, for the sole purpose of training a hand-pose estimator model, a ground truth (GT) center-of-mass point can instead be chosen as one of the annotated hand joints. This is the MCP joint (lowest joint of the middle-finger) as seen in the official implementation of Deep-Prior++. However, when reporting results, all recent works make use of some form of hand localisation methods and thus reporting hand-pose estimation results with the use of GT annotations for cropping images that are then used to train a model may cause inconsistency when comparing

other recent work although the difference won't be very significant (as hand detection is moderate clutter is considered a solved problem). This was also observed in [18].

Deep-Prior++ further improves on hand localisation from thresholding-based center-of-mass (CoM) prediction using a relatively shallow convolutional network. The initial cropped image is fed into this network which then outputs a 3D offset for the MCP joint, which as described above is often considered the reference point for GT CoM label. Since the MCP joint is considered as the CoM point, the offset value is used to then refine previous estimated of the CoM and thus get a better crop of the hand. This network is separate to the two-part model discussed earlier. In practice, it makes a significant boost in hand-localisation however the overall hand-pose estimation in only marginal (1mm improvement in one of the data-sets used during original experiments [18])

Regardless of our final model choice for estimating hand-poses we will ensure that final comparative results are reported based on center-of-mass points found using one or more hand detection techniques described above. Furthermore, we will have a much broader practical applicability for our whole framework if it is suitable for a real-time environment. Although this is not a requirement in any way of this project, in order to demonstrate any real-time pose-estimation, reliable hand localisation techniques must be incorporated.

2.2.3 Pictorial Structure Model

A pictorial structure (PS) model is an undirected graph representation of distinct parts of a generalised structure. Ignoring our notation in 1.3.1 for this particular sub-section, the graph representation can be described as $\mathcal{G}(\mathcal{V}, \mathcal{E})$. With the set of vertices in $\mathcal{V}$ corresponding to each part in the model's structure and edges in $\mathcal{E}$ corresponding to the unique pairings between different parts. $l_i \in L$ correspond to the location of each i^{th} joint and equivalently the value of each i^{th} vertex. $e_i \in E$ describes the k^{th} pair tuple as (l_i, l_j)

Generally speaking, in a PS model, l is optimised using mean-square error (MSE) or other task appropriate loss functions that describe the difference between the output approximated by the model and the ground truth. The 'mis-match loss' $m(\cdot)$ is used to describe such a function. For measuring the discrepancy between $e_{predicted}$ and $e_{kinematic}$ i.e. the predicted and kinematically defined part-pairs, a 'deformation loss' $d(\cdot)$ is used.

General Representation

$$E = (e_1, ..., e_K)$$
$$L = (l_1, ..., l_n)$$
$$e_k \equiv (l_i, l_j);\ i \neq j$$
$$\forall\, 0 \leq k \leq K, 0 \leq i \leq n;\ 0 \leq j \leq n$$
$$m : L, I \rightarrow \mathbb{R}$$
$$d : E, I \rightarrow \mathbb{R}$$

$$L^* = \arg\min_{L}(\sum_{i=1}^{n} m_i(l_i) + \sum_{k=1}^{K} d_k(e_k)) \tag{2.2}$$

Many frameworks have been proposed for optimising pictorial structures with the first one introduced in 1973 [39] . The PS model is particularly applicable to feature localisation in objects of interest and in objects involving the human body as many strong assumptions can be made on the body's general representation.

A PS model for the whole or partial human body can be described using locations of sub-parts along with their relationships with each other [51]. Well studied kinematic constraints of the human body can be used to model the variations (or lack of) in the relationships between these sub-parts to ensure any final predictions are actually humanly possible. Alternatively these constraints can be learnt by fitting a statistical model over the training data-set.

$$\mathcal{M} : (\mathcal{S}, \mathcal{P})$$

$\mathcal{S}$: set of possible observations i.e. sample space

$\mathcal{P}$: set of probability distributions over $\mathcal{S}$

$$\mathcal{P} = \{P_\phi : \phi \in \Phi\}$$

In practice, Guassian Mixture Models (GMMs) are a popular choice for describing P_ϕ as they allow the modelling of each joint-pair as a separate normal distribution. Based on the specific task, some researchers simplify E to only consider those pairs that directly connected to each other which then results in a undirected but much simplified graph in this case $|E| \neq n$, otherwise and for the equations described below, $|E| = n$.

Optimising 2.2 is usually realised using a MAP (maximum a posteriori probability) estimate of L. The MAP estimation makes use of unaries and binaries. A unary is defined as a function that produces a probability score based only on a particular joint's location i.e. l_i whereas the binary is a probability score function based on joint pairs i.e. e_k.

A 'binary potential' $p(L|\theta)$ is a prior learnt from the training set which probabilistically answers the 'realism' of the pair vectors defined in E given a learnt kinematic constraint model with parameters contained in θ as shown later. This is done by calculating the likelihood that a certain relative difference vector $l_i - l_j$ is from the same distribution as the relative vectors present in the training data-set as a probability score $[0, 1]$. It is said to be 'binary' as it requires a tuple of joint vectors (l_i, l_j) as input.

On the other hand, the 'unary potential' $p(I|L, \theta)$ asks the likelihood [0, 1] of a joint j being in a particular location x_j on a 2D plane. For this we can have many methods (e.g. heatmap pixel-wise score) but in this paper convolutional layers were used to extract patches and then use a set of random forests, one for each patch for each joint location. A particular leaf node in each tree of every forest predicts

It makes sense that the unaries depend on the input image by the binaries don't as intuitively binaries use relative vectors between joints and thus translational invariant and can be modelled as gaussian mixture models over the dataset whereas the unaries use exact location vectors which would largely vary form image to image.

θ can be be modelled as a tuple containing appearance parameters u_i, connection parameters c_k and what pairs of vertices are present in the kinematic model as $\mathcal{E}$. u_i describes the representation of features around the i^{th} joint for e.g. cropping a fixed area around the i^{th} joint and then passing it through a CNN. c_k represents the parameters of a statistical model fitted over the dataset for the i^{th} joint and is equivalent to ϕ described earlier

$$\theta = (u, \mathcal{E}, c)$$

For unary optimisation, c is irrelevant and can be considered a constant. For binary optimisation, u is irrelevant and can be considered a constant. $\mathcal{E}$ is always constant so can be ignored for either optimisation cases. If we use deep-learning based unary optimisation, u_i can often be reduced to u i.e. the whole image is used to describe features around a particular joint (no need for cropping). Also note that $p(l_i, \theta)$ i.e. the absolute position must not have any constraints based on priors so is considered constant for optimisation.

We can show that the MAP estimation 2.3 of a PS model is equivalent to minimising the cost function described in 2.2 [51]:

$$p(I|L,\theta) = p(I|L,u) \propto \prod_{i=1}^{n} p(I, l_i, u_i)$$

$$p(L,\theta) = f(p(l_i, l_j|\theta), p(l_i,\theta)) \propto \prod_{(l_i,l_j)\in E} p(l_i, l_j, u_i|\theta)$$

$$p(L,\theta) \propto \prod_{e_k \in E} p(e_k, u_i|\theta)$$

$$p(L,\theta) \propto \prod_{k=1}^{K} p(e_k, u_i|\theta)$$

$$p(L|I,\theta) = \frac{p(I|L,\theta)\cdot p(L|\theta)}{p(I|\theta)}$$

$$p(L|I,\theta) \propto p(I|L,\theta)\cdot p(L|\theta)$$

$$P(L|I,\theta) \propto (\prod_{i=1}^{n} p(I, l_i, u_i) \cdot \prod_{k=1}^{K} p(e_k, u_i|\theta))$$

$$L^* = \arg\max_L (\prod_{i=1}^{n} p(I, l_i, u_i) \cdot \prod_{k=1}^{K} p(e_k, u_i|\theta))$$

$$L^* = \arg\max_L (\sum_{i=1}^{n} \log(p(I, l_i, u_i)) + \sum_{k=1}^{K} \log(p(e_k, u_i|\theta)))$$

$$L^* = \arg\min_L (\sum_{i=1}^{n} -\log(p(I, l_i, u_i)) + \sum_{k=1}^{K} -\log(p(e_k, u_i|\theta))) \tag{2.3}$$

2.3 Human Action Recognition

2.3.1 Overview

Human action recognition is a term used for any model which has an objective of classifying actions that have taken place in a sequence of input frames. The input frames usually have an RGB modality as they originate from video clips. These actions can either occur for the entire duration of the recording or only start/stop within a specific time-frame. Sequence-based inputs (e.g. videos) require the efficient feature extraction of an additional paradigm on top of spatial information, namely the complex relationship between each successive frames which is known as the temporal information. At the beginning of this decade video action recognition involved mainly the use of RNN-based networks. However, with the introduction and subsequent popularity of optical-flow based two-stream CNNs [52] for video action recognition, many new CNN-based architectures have been proposed in research and most recent top performing solutions are based on CNNs.

Typical challenges in action recognition include the huge computation costs for training videos that scale with not just input image resolution but also the video duration as well as the problem of capturing important contexts over a longer period of time while forgetting irrelevant ones in video clips [53].

Newer CNNs based approaches have merged the two-stream method into one by making use of 3D-CNNs [54]. Just like with hand pose estimation, 3D conventional filters enable the capturing of data in an additional dimension which in the case of video-based action recognition is the temporal dimension.

Although there are many varying approaches to video action recognition (some variants shown in 2.2), there hasn't been as many contributions towards 3D skeleton based action recognition. Though most of the approaches described for video action recognition may not be appropriate for skeleton-based action recognition, we can still gain some insights on what architectures may work and provide a boost in performance. Since our objectives mean our focus is mainly on skeleton-based action recognition, we do not study the different approaches to video-based action recognition in detail and refer the reader to [55, 53] for an overview of recent popular approaches in video-based action recognition.

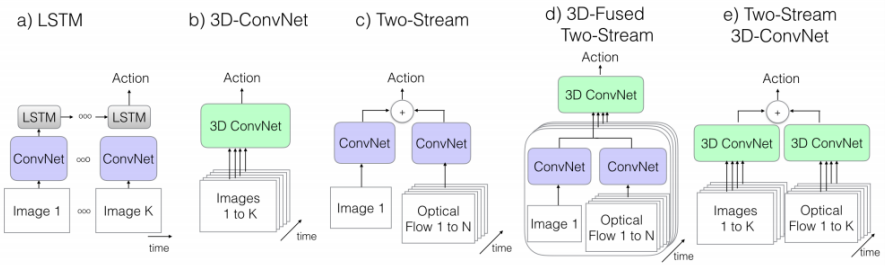


Figure 2.2: An architectural overview of the main themes currently employed in video-based action recognition. Image Source: [56]

However, in order to compare our model with existing techniques, we will require a comparison with depth-based action recognisers because that is the modality for inference. This means that we should be comparing against previous works on egocentric depth-based (or RGB-based with input channels reduced to 1) activity recognition . Most of these works have perform measures on other data-sets than what we propose to use, these data-sets may or may not include hand-pose annotations thus the only way to successfully compare is to train the model on our data-set and report the performance. To stay within the time scope limitations of this project we will keep this task as a lower priority until the main objectives of this project are achieved. For a fairer comparison with our baseline skeleton-based action recognition, we will make use of a CNN-RNN architecture around the same time frame as the skeleton-based model (see 2.3.3 below). A good candidate for such an architecture is described in [57].

2.3.2 Relevant Branches

Skeleton Based Action Recognition

There have been several proposed approaches designed for skeleton-based action recognition [58, 59, 60, 61, 62, 63, 64, 65]. However many of these approaches involve first extracting pose-information from videos and then using another architecture to perform pose-based action recognition. Nevertheless the second stage of these approaches give useful insights on the various techniques that could be employed for pose-based action recognition. An interesting work is in [65] where pose probability heat-maps are aggregated (based on soft-max at each frame) to represent a colourised or pictorial 'flow' of each joint for the entire sequence in a compact RGB image form. These RGB features can now be passed through a standard or state-of-art 2D classifiers for action recognition, completely by-passing the need for RNN, two-stream 2D-CNN or 3D-CNN based models for action recognition. Furthermore there has been a lot of work in recent times across many domains over attention-based networks which seek to improve networks inspired from a biological perspective of selective 'attention' or gaze in humans [66, 67]. In light of this, many approaches to action recognition deploy some form of attention mechanism based on poses [68, 69, 63]. We explore the general notion of attention mechanism in the context of translational networks [70] in 2.3.4. For up-to-date information on state-of-art approaches to skeleton-based activity recognition (including hands and body) please refer to [71, 72].

Egocentric Action Recognition

Contributions have also been made towards egocentric activity recognition which is what our project comes comes under. In [73] this paper the use of specific image transformer modules are explored that help clear global camera motion due to the fact that egocentric videos are recorded in first person view and thus is most likely to be worn by a person and hence appear shaky.

Contribution towards better action recognition using hands as visual cues is seen in [74]. Instead of using depth-maps, a hand mask encoder first converts a set of RGB hand images from an action sequence to segmented masks and then these are fed to an RNN. Our approach will be somewhat similar to [74] as we would most likely perform background removal and cropping before performing passing a depth image through a hand pose estimator or generator. Note we need to do this a 'pre-processing' step rather than a learnt step because unlike [74], our choice of data-set [16] contains a lot more severe occlusions and interactions of hands with other objects and thus we leave the task of segmentation to a different framework.

In [75], specific attention modules are used to improve egocentric action recognition.

2.3.3 Co-occurrence Feature Learning: A Baseline

For simplicity, most diagrams describing LSTMs depict single-unit LSTMs which may wrongly imply that a large dimensional input such CNN-based features or one-hot vector encodings are all transformed to 1D when passed through an LSTM layer. In practice, this doesn't happen and the 'LSTM layer' is formed of H hidden LSTM units just like the neurons of a FC layer giving an intermediate hidden output vector as $\boldsymbol{h} \in \mathbb{R}^H$.

As a baseline we make use of co-occurrence learning of joints for LSTM-based action recognition [58]. To summaries, co-occurrences measure how closely some joints are linked with respect to others by weighting the joint's positions appropriately. This notion can be modelled using a standard representation of LSTMs i.e. each layer represented by $\boldsymbol{h}$. This implies that the baseline can be realised by simply defining one or more vanilla LSTM layer(s) for our network with H hidden units (the dimension of $\boldsymbol{h}$) in each layer (note H is not required to be fixed across layers).

Following the evaluations performed by [16], we will make use of only a single hidden layer $\boldsymbol{h}$ to encode temporal information (i.e. the hidden states at various time-steps) present in a video sequence. This can be described in a matrix form with each row representing the value in the hidden state at time t:

$$\boldsymbol{H} = [\boldsymbol{h}_1^T, ..., \boldsymbol{h}_T^T]$$

For the final output, a FC layer is followed by soft-max activation ($\sigma(\cdot)$) to predict the final video class. The FC layer is required because usually $H \neq C$.

$$\boldsymbol{w} = \sigma(\boldsymbol{M}\boldsymbol{h} + b) \tag{2.4}$$

Here $\boldsymbol{w}$ is the class probability vector following our notation in 1.3.1, $\boldsymbol{M}$ and b are learned parameters of the last fully connected layer to encode (project) $\boldsymbol{h}_T$ into a dimension consistent with $\boldsymbol{w}$ i.e. $\mathbb{R}^C$

2.3.4 Attention-Based Models

This sub-section introduces attention-based models specifically within the context of RNN-based translational models. It then briefly explores the use of attention modules in other scenarios. We do not follow any notation from 1.3.1 in this sub-section.

Traditional RNN-Based Translation Networks

Usually when we perform translation of a sentence, we would first take an entire sentence, break it into words, then transform each text-word to a numeric encoding in the word-space $w_o \in \mathcal{W}_{orig} \subseteq \mathbb{R}$ giving a sentence space vector $\boldsymbol{s} \in \mathcal{S} \subseteq \mathbb{R}^{\mathrm{N}}$ (sentence embedding) for N words [70].

$$w_o \in \mathcal{W}_{orig}$$

$$\mathbf{s}_i = [w_{o1}, w_{o2}, ..., w_{oN}]^T$$

Then we simply convert this using:

$$f : \mathcal{S}_{orig} \rightarrow \mathcal{L}_{orig} \subseteq \mathbb{R}^{\mathrm{L}}$$

Where $f(\mathbf{s})$ is a vector mapping function that word-by-word (starting from the first word) computes the translated embedding representation $\mathbf{l} \in \mathcal{L}_{orig}$.

$$\boldsymbol{l}_{curr} = g(w_{o_{curr}}, \boldsymbol{l}_{prev})$$

$$g : \mathcal{W}_{orig} \times \mathcal{L}_{orig} \rightarrow \mathcal{L}_{orig}$$

Note: the embedding representation $\boldsymbol{l}_i$ is a single vector taking into account all information presented in the sentence up until position i (starting from 1) in the sentence. This is usually done via a LSTM cell in the encoder where for each position, the latent vector (current state) is updated using previous state representation $\boldsymbol{l}_{prev}$ and the current word's word-space representation w_{curr}. In the end we get a single vector $\boldsymbol{l}_{final}$ to be fed to decoder.

To instantiate decoding (translation), first $\boldsymbol{l}_{final}$ is transformed to $\boldsymbol{t}_{initial} \in \mathcal{L}_{trans}$ which is the vector representation in the translated space using a single-step vector function $m(\cdot)$.

$$m : \mathcal{L}_{orig} \rightarrow \mathcal{L}_{trans}$$

Now, each word-by-word translation (where) is performed by:

$$\begin{aligned}
&w_t \in \mathcal{W}_{trans} \\
&w_{t2} = h_1(w_{t1}, \boldsymbol{t}_1) : \text{LSTM output at } i = 2 \\
&\boldsymbol{t}_2 = h_2(w_{t1}, \boldsymbol{t}_1) : \text{LSTM state at } i = 2
\end{aligned}$$

$$\begin{aligned}
&h_1 : \mathcal{W}_{trans} \times \mathcal{L}_{trans} \rightarrow \mathcal{W}_{trans} \\
&h_2 : \mathcal{W}_{trans} \times \mathcal{L}_{trans} \rightarrow \mathcal{L}_{trans}
\end{aligned}$$

i.e. h_1 maps an (previous translated word, previous state) pair to the next translated word and h_2 maps the same pair to the next state.

Incorporating Attention

Our brain in general doesn't do word-by-word translation by first storing all information of the original sentence in latent-space, instead we tend to break sentences down and translate it phrase-by-phrase while other parts are untranslated.

Attention module in simple terms works by looking at all hidden states of the original sentence rather than just the final one i.e. in our case:

$$\boldsymbol{L} = [\boldsymbol{l}_1^T, ..., \boldsymbol{l}_N^T] \in \mathbb{R}^{N \times L} \text{ where } \boldsymbol{l}_N^T \equiv \boldsymbol{l}_{final}$$

$\boldsymbol{L}$ is a collection of vectors (as a matrix) with each i^{th} row-vector corresponding to the hidden state of the original latent space for sentence position i. The attention module then tries to guess which of these states (or equivalently words) would be relevant to produce the i^{th} translated word i.e. what positions to 'pay attention to' when looking to translate the i^{th} word in the original sentence. One would expect positions around the i^{th} word having high probability of attention.

First, $\forall 1 \leq j \leq N$, function $m_1(\cdot)$ calculates the energy $e_j \in \mathcal{E}_i$ to assign to each input hidden state $\boldsymbol{l}_j \in \mathcal{L}_{orig}$ given the current most recent translated state $\mathbf{t_{i-1}} \in \mathcal{L}_{trans}$. Notice how the energy-space will be different for each i^{th} position in the sentence. In the original paper [70] this was done using feed-forward neural network based weighted summation of $\boldsymbol{l}_j$ and $\boldsymbol{t}_i$ with the weights shared $\forall j$.

$$m_1 : \mathcal{L}_{orig} \times \mathcal{L}_{trans} \to \mathcal{E}_i$$

Then, $\forall 1 \leq j \leq N$, function $m_2(\cdot)$ computes probabilities as a 'soft-max' activation procedure using each of the energies as input such that $\sum_{j=1}^{N} p_j = 1$ where $p_j \in \mathcal{P}_i$. This can be thought of a simple activation function applied to all nodes in a layer exactly the same as last-layer activation of multi-class classification.

$$m_2 : \mathcal{E}_i \to \mathcal{P}_i$$

Lastly, all such probabilities and ordered collection of latent states $\boldsymbol{L}_{orig}$ are combined to give a single context vector $\boldsymbol{c}_i \in \mathcal{C}_{trans} \subseteq \mathbb{R}^{\mathrm{L}}$ using $m_3(\boldsymbol{p}_i, \boldsymbol{L}_{orig}) = \boldsymbol{L}_{orig}^T \boldsymbol{p}_i$ where $\boldsymbol{p}_i = [p_{i1}, \ldots, p_{iN}]^T$.

$$m_3 : (\mathcal{L}_{orig} \times \mathcal{P}_i) \times \cdots \times (\mathcal{L}_{orig} \times \mathcal{P}_i) \to \mathcal{C}_{trans}$$

This last operation can be thought of doing the expectation over the different latent states as its a weighted mean. The idea is to choose the best state that would, on average, be the most relevant or 'requires the attention of' from the input sentence (now converted to a set of latent states) given the previous output state. This is opposed to just looking at a representation of all previous states which is inferred from $\boldsymbol{l}_{orig}$ only once as shown in the previous section.

The new word-by-word translation scheme is changed to:

$$\begin{aligned}
c_2 &= m_3(\boldsymbol{p}_1, \boldsymbol{L}_{orig}) \\
w_{t2} &= h_1(w_{t1}, \boldsymbol{t}_1, \boldsymbol{c}_2) \\
\boldsymbol{t}_2 &= h_2(w_{t1}, \boldsymbol{t}_1, \boldsymbol{c}_2)
\end{aligned}$$

$$\begin{aligned}
h_1 &: \mathcal{W}_{trans} \times \mathcal{L}_{trans} \times \mathcal{C}_{trans} \to \mathcal{W}_{trans} \\
h_2 &: \mathcal{W}_{trans} \times \mathcal{L}_{trans} \times \mathcal{C}_{trans} \to \mathcal{L}_{trans}
\end{aligned}$$

CNN-RNN Based Video Activity Recognition

Specifically for text translation models we are using RNNs to encode words to latent space such that the features not just represent the current word but also what words have been seen before. In bi-direction RNNs, this is a step further to also include future states in the current latent vector.

In the case of video action recognition, we can first encode frames to rich feature vectors to learn spatial info using 2D-CNNs and then later use these to produce state vectors to learn temporal information much similar to latent vectors in text translation models.

CNN-Attention Based Video Activity Recognition

The attention concept described above can be directly applied to '3DCONV' filters where not just a patch of input image but also some set of image sequence frames from a video are used to calculate the next activation. Here we can use a framework called CBAM i.e. convolution-based-attention-module [76]. CBAM produces probability scores to choose the best 'channel' at each layer and another set of scores to choose the best 'locations' for a particular given input. This means that given an input and thus previously learned features of such input the model learns to produce probability scores similar to the $\boldsymbol{p}_i$ vector to element-wise multiply a constant for each channel and a constant for each location in all channels. As a result, both temporally and spatially, the most important features are selected or 'given importance to' rather than ranking the whole 3D volume equally as with 3D-CNNs.

2.4 Other Related Works

Refinement using Action Class Probabilities

This work [77] makes use of techniques described from the PS model framework described in 2.2.3. In this paper, human skeleton (pose) is first predicted as a crude estimation with the belief that all action classes are equiprobable. A sequence of these are then used to produce an action class probability vector whose components sum to 1.

Next, this vector is used as a new belief to update the prior which is the original crude estimation. In particular the unaries and binaries (See 2.2.3) are now additionally conditioned on the action class probability vector. An important insight of this paper is that a performance gain for binaries with the use of conditioning is achieved by thresholding the most probable class to 1 and the rest to 0 rather than using the soft-maxed output. However, for unaries, the action class probability vector is actually better than the thresholded version.

Unfortunately, we cannot however, use this technique directly for our hand-pose estimation problem due to the fact that u_i is required to be modelled using patches extracted from a pre-trained body part detector [78] as mentioned in the original paper [77]. Even if we train such network, extracting accurate patches of hand joints is a much more complex task as compared to body parts and especially so in the depth domain because (a) hand joints are much close to each other than human skeleton joints and (b) several joints cannot be distinguished from each other when cropped separately due to their visual similarity. This is even so harder in the depth domain.

Thus, we conclude that while some ideas form this paper, namely the unary and binary potential maximisation, can be implemented, due to the reasons explained above the complete model architecture as it stands cannot be implemented for the hand domain and thus we can't empirically compare this approach on a hand activity data-set.

The biggest insight of this paper is the successful use of the concept of updating soft-max action class probabilities in a looped fashion and this formes a basis for our proposed model.

Refinement using Privileged Information

This paper [15] approached the problem of hand pose estimation and action recognition by imposing constraints on depth-encoding to match a latent-space of hand poses. The network then uses a multi-task loss to minimise (via RNN states) the action classification loss and in parallel using the the same RNN state space, a hand-pose estimation regression loss is minimised. The 'privileged information' in this context is the ground-truth pose information for the refinement training. This methodology is also a special case of the LUPI framework [13]. This encourages the model to produce RNN states that contain both class information and hand pose information (which is then later separated using separete FC layers). In this paper hand pose estimation is further refined using a expectation maximisation approach for the iterative refinement of the model.

Our training procedure will differ from this model in several ways. Firstly, we do not aim to estimate hand-pose and action class probability at the same time which is the usual case in multi-task learning. Instead, we consider a staged prediction approach as this way we can independently experiment with different hand pose estimators or generators to find out the most suitable sub-model and as a result our framework can be more generic and adaptable. Secondly, we consider the use of cVAE instead of EM (Energy Minimisation) for the refinement step. There can be several

reasons for and against this approach and as a fall-back the EM approach can be explored if cVAE doesn't produce any fruitful results.

Unlike [77], this model can actually be used as a good baseline for comparison. An important point to note is that the action class probability vector is not refined and only the first prediction is used for classifying actions.

End-to-End Multitask Learning

Another interesting approach to the problem of estimating both hand-poses and recognising actions in (third-person) RGB video clips is to employ a multi-task learning framework in an end-to-end (deep) fashion [79]. This is a great unification of methods that tend to first extract probability heat-maps, use them to estimate pose information and then make use of the given poses as features for training an video action recognizer. The authors of this paper successfully showed that by replacing argmax (to recover pose coordinates) with its differentiable variant soft-argmax to recover pose co-ordinates and enable end-to-end optimisation of both pose estimator and action recognizer. This method's end goals are very similar to our approach and due to the end-to-end training procedure and the fact that the action recognizer uses features form different modalities, it might provide better results that our proposed model. However, the training time and complexity of such a model can be huge. Lastly, this model doesn't showcase the successful use of action information to improve pose estimation. Thus at-least, on the pose estimation aspect, we can imagine our proposed model to perform better. Nevertheless this can be a strong alternative model to compare our approach from.

Chapter 3

Implementation Plan

3.1 Data-set

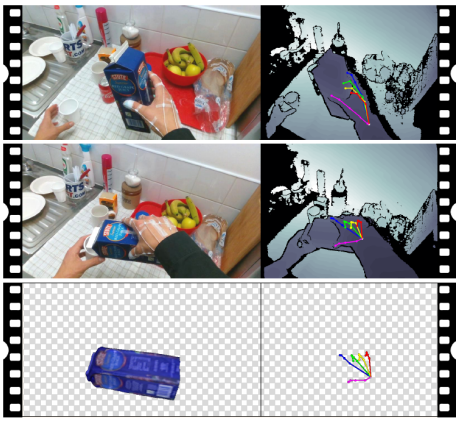


Figure 3.1: A showcase of the different modalities of data present in our chosen dataset. Image Source: [16]

The data-set we have chosen for training and testing our proposed models is described in [16]. It consists of 45 classes of wide variety of everyday actions. It is the most recent and detailed data-set of its kind with annotated information on several modalities: RGB (top-left), depth (top-right), action labels, selected object 6D poses (bottom-left) and 3D hand poses (bottom-right). Our concern is to make use of annotated depth, 3D hand pose and action labels for training a network that can later predict both hand poses and action labels using just depth information. For this scenario the data-set seems very appropriate.

There is a large variation in the number of frames for different sequences as it is mentioned that some actions are a lot slower than others. This also introduces a variation aspect in the data-set and matches a lot closer to real-world action where some are a lot more straight forward than others. Due to this large variation we will be restricted on models that accept variable length sequences for action recognition. Fortunately, our chosen baseline is capable of achieving that.

The average video sequences per class in this data-set is about 26. This seems to be a reasonable amount but there is a good risk that any deep-learning we employ over-fits on the training set. As a result we would need to be careful at monitoring over-fitting and if required use appropriate regularisation approaches. Please refer to [16] for a detailed taxonomy of the data-set.

3.1.1 Baseline Models

In order to empirically demonstrate the usefulness of our model, we must compare it against state-of-art methods. This is a somewhat harder problem than usual because very recent models have not yet been evaluated using our data-set and furthermore there exists a very few models that too a big extent have demonstrated what we are trying to achieve (i.e. the refinement of multi-task

prediction) and these were released before the data-set was published so may be inappropriate for comparison.

Nevertheless, the individual models in our framework, namely the HPG (Hand Pose Generator) and HAR (Hand Action Recognizer) can be relatively easily compared with other models of the past. Furthermore, we can choose a recent baseline to progress our development. This can be any model that has a straight-forward implementation and which can be easily modified if required to showcase ablation studies or to make modifications.

Hand Pose Estimator (HPE)

As described in section 2.2.2, we will be using Deep-Prior++ [18] as a strong baseline for comparison of our model. We can also use this model as a basis for the architecture of our HPG model. For e.g. designing an auto-encoder framework for both the depth and pose modality can be based on the structure of this model (e.g. the convolutional filters). Furthermore, this method can be used as a fall-back should things not go as planned and we decide to pursue hand pose estimators rather than generators for our pose-estimation model. This model also involves many pre-processing steps which will be helpful given the naturally huge variation of hand poses in our choice of data-set. We have already implemented most of the features of this model and will soon begin testing its performance on the chosen dataset.

Hand Action Recogniser (HAR)

$$g : \mathcal{V} \rightarrow \mathcal{W}$$

This model forms a basis for predicting action class labels given a sequence of hand-poses as described in 2.3.3. A single layer LSTM network will be used as our baseline. Even though this LSTM network won't produce results on-par with recent advancements, it achieves reasonable results on the given data-set and its simplicity means that it will be easy to implement, train and extend to our proposed framework. Furthermore, if time persists we can look at implementing near state-of-art skeleton-based acton recognizers to investigate performance improvements for using a more complex network. However this is left as an optional task.

Multi-modal Hand Pose and Depth Generator (HPG)

$$f : \mathcal{U} \rightarrow \mathcal{V}$$
$$h : \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{V}$$

We will study the use of probabilistic mapping which produces parameters of a posterior conditional probability distribution that can be used to sample multiple poses given a single input depth-map. This will be helpful in the context of multi-instance learning or simply for the purpose of data-augmentation. The function f signifies the mapping when the action class is a uniform prior and the function h signifies a mapping when the action information is known (estimated) to some extent and the same depth information is used to refine a hand-pose estimate.

Several recent contributions have been made towards the possibility of cross-modal data generation as a possible solution for the task of hand-pose estimation[21, 34, 80, 81]. The high-level idea behind all such contributions is to make use of generative models to learn a shared latent space [80] or one or more transformation functions [21]. This rich latent space can then be used to transform any data from one modality to the other. Most such works have shown that the results are on-par or slightly worse than recent advancements in deep discriminative models for the purpose of hand pose regression.

Furthermore, the added benefit of using generative models is the possibility of performing unsupervised or weakly-supervised (partial ground truth data and mostly synthetic data) training of any auxiliary hand pose estimator (HPE) models which could then possibly alleviate generalisation issues incurred in such models [22].

Although many complex techniques have been proposed for the purpose of cross-modal data generation, as a first step towards this area, we propose a baseline architecture similar to [80] with the only difference being that we plan on using a conditional-VAE (cVAE) [82] instead of a VAE with the action-label imposed as a condition.

Details on the various ablation studies on the VAE-based cross-modal learning framework is described in detail in [80]. It is however worth mentioning that the main focus of that study was RGB to hand skeleton cross-modality. However, the work showed sound (albeit not state-of-art) results for depth to hand skeleton estimation. In essence, to facilitate cross-modal latent space learning, two (Encoder, Decoder) pairs are used which can be represented as two separate VAE models.

The study concluded that the overall best performing variant of the generic VAE-based cross-modality architecture was when both RGB-RGB and RGB-Depth reconstruction losses were used to update the weights on the corresponding auto-encoders. This can be practically applied by choosing between computing $\tilde{v}$ (joints) or $\tilde{u}$ (depth-maps) by either splitting the training data-set into two portions or randomly choosing between the two computations for each mini-batch.

When $\tilde{u}$ is computed, the back-propagated errors can be used to train VAE_u in the usual way, however when $\tilde{v}$ is computed, only the weights of Enc_u and Dec_v are updated on back-propagation. This is summarised by the equations below. Apart from the usual VAE terminology we introduce $l \in \mathcal{L}$ as a one-hot encoded action label prior for the formulation of a conditional-VAE (cVAE) [82]

We follow our convention of using u for depth-maps and v for 3D hand-joints in section 1.3.1 but disregard the distinction between scalar u, vector $\boldsymbol{u}$, matrix $\boldsymbol{U}$ or tensor $\mathbf{U}$ for brevity. The algorithm is adapted from [80]. The two models are alternatively trained (based on what mode is selected for the current training iteration) as:

$$\begin{aligned} p_u \equiv Enc_u : (u, l) \rightarrow z \sim \mathcal{N}(\mu_{u,l}, \sigma_{u,l}) \\ p_v \equiv Enc_v : (v, l) \rightarrow z \sim \mathcal{N}(\mu_{v,l}, \sigma_{v,l}) \\ q_u \equiv Dec_u : (z, l) \rightarrow \tilde{u} \\ q_v \equiv Dec_v : (z, l) \rightarrow \tilde{v} \end{aligned} \tag{3.1}$$

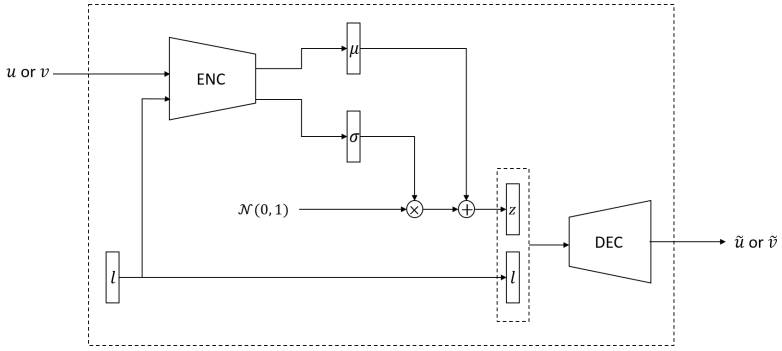
$$\begin{aligned} \tilde{u} = (q_u \circ p_u)(u) \\ \tilde{v} = (q_v \circ p_u)(u) \end{aligned} \tag{3.2}$$

$$\begin{aligned} \ell_{MSE_u} = ||u - \tilde{u}||_2 \\ \ell_{MSE_v} = ||v - \tilde{v}||_2 \\ \ell_{KL} = \mathcal{D}_{KL}(\mathcal{N}(0,1)||\mathcal{N}(\mu_{u,l}, \sigma_{u,l})) \end{aligned} \tag{3.3}$$

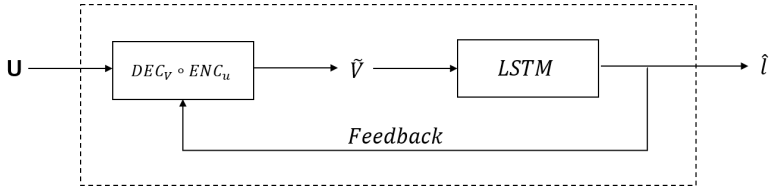
$$\begin{aligned} \ell_{p_u,q_u} = \ell_{MSE_u} + \ell_{KL} \\ \ell_{p_u,q_v} = \ell_{MSE_v} + \ell_{KL} \end{aligned}$$

Apart from the losses mentioned in [80], similar works have made use of a 'cyclic consistency loss' that runs through a loop of transformations to get data back in the same modality as the input[15]. Although this is much suited for GANs or GANs+VAEs, We can still experiment with a loss term for the cyclic consistency in VAEs. After Calculating $\hat{u}$ as the final modality (depth), standard MSE reconstruction loss can be used for this loss as follows:

$$\begin{aligned} \hat{u} = (q_u \circ p_v \circ q_v \circ p_u)(u) \\ \ell_{MSE} = ||u - \hat{u}||_2 \end{aligned} \tag{3.4}$$



(a) cVAE



(b) Overall

Figure 3.2: (a) cVAE Architecture for both p_u, q_u and p_v, q_v, (b) overall block diagram with depth-frame sequence tensor $\mathbf{U}$ as input, matrix $\tilde{\boldsymbol{V}}$ as the interim output consisting of joint sequence row vectors and $\hat{\boldsymbol{l}}$ as soft-maxed action class probability vector.

Implementation Diagram

The implementation diagram can be found in Figure 3.2.

Extensions

Once the standard baseline framework as described in section 3.1.1 is realised and works as expected (or have some unexpected concerns) we can experiment with various promising approaches to enhance the model and at the same time help us better understand and empirically verify what approaches are better to pursue.

Some approaches in this regard include:

- Gaussian Process Approximation using dropouts at test time [11] to produce multiple samples based on an underlying distribution of the weights in a network.
- Modifying the HAR architecture to incorporate multi-instance learning based on multiple probable candidate samples from HPG for each input depth-map [83].
- Modification to the loss functions used for HAR to improve temporal consistency.
- Exploring state-of-art approaches under the 'multi-task learning' and 'privileged learning' [13] paradigm.

3.2 Evaluation

There are many approaches to evaluating our proposed model and since its based on a multi-task learning paradigm there can be several kinds of models that we can compare with. However, to prevent scope creeps in the project we will stick to comparisons with simple to implement, train and publicly available models that have been shown to work with our modalities.

Ground Truth Pose Annotations

The first basic evaluation to perform would be to compare performance of proposed changes again an absolute performance limit that could be achieved when using the same HPG architecture along with ground truth samples to train it. Theoretically we can't beat this performance but would

expect a good framework to not deviate too far away from this score. The score will mainly be affected by our choice of HPG/HPE and how we model the feedback loop. It will be interesting to find out if we can beat that score which is a possibility if we consider the grond truth annotations to be noisy which is often the case in practical data-sets.

Ground Truth Action vs No Action vs Uniform Action Annotation

Another aspect to consider would be answering the following questions:

- Is conditioning the HPG with action information actually helpful to estimate poses from depth as compared to directly using depth information? Is this information a weak or meaningless prior?
- How well does our feed-back based approach work in comparison to (a) the first estimate? and (b) ground truth action class probability vector (i.e. one-hot)?

The first-point is an important aspect to consider and if we find that conditioning a HPG on action makes worse prediction than the same like-for-like architecture (except for the conditioning aspect) than we may have to consider other areas for further investigations. This will form as a first milestone that we wish to achieve once a baseline HPG has been prototypes.

Chapter 4

Roadmap

4.1 Progress Milestones

Milestone	Due Date	Status
Baseline HPE Model Port	18-Dec-18	Completed
HPE Training Pipeline Port	05-Dec-18	Completed
HPE Data Augmentation Port	23-Jan-18	PCA + Scaling Augmentation Left
Evaluate HPE Performance	05-Feb-18	
Baseline LSTM Action Recogniser	05-Feb-18	
Finalise HPE Port (Buffer)	10-Feb-18	
Finalise HPE & HAR Baseline Scores	14-Feb-18	
Implement cVAE-Feedback Model	25-Feb-18	
Experiment with another HPG variant (e.g. GAN-based)	28-Feb-18	
Finalise HPG (Buffer)	10-Mar-18	
Implement Fusion and/or Multi-Task Methods	22-Mar-18	
Implement Further Novel Exp. or Fall-back Methods	20-May-18	
Finalise Best Working Prototype	01-June-18	

Table 4.1: A set of indicative tasks to evaluate progress and seek fall-back approaches if required. Note: the large gap between 22th March and 20th May is because most of this period is reserved primarily for exams.

4.1.1 Fall-back Approaches

1. The use of GAN+VAE or only GAN based cross-modality latent space learning (Low). [15, 21]
2. The use of EM (Energy Maximisation) procedure instead of cVAE (Moderate). [15]
3. The use of pictorial structure model as described in [77] (Moderate).
4. Only considering improvements for either action-recognition or hand-pose estimation (Severe).
5. Only considering baseline models for hand-pose estimation and action-recognition and only experimenting with the use of drop-outs at test time i.e. scraping feedback idea (Severe).

Since we have not yet observed the use of conditional variational auto-encoders in the context of conditioning actions for producing better hand-pose estimates, this is an area of novelty and as such carries a good amount of risk. Having understood that, we have proposed above a set of fall-back approaches which have already been well-tested, some to a greater extent than others,

to show their usefulness. Although theoretically our model is simple and doesn't have many underlying assumptions, it may not work due to several unpredictable reasons that might arise during development. The fall-back approaches are listed in the order of severity and based on what aspects of our proposals do work we would select one of these possible options. We hope that this is not the case and we are able to deliver at-least the main objectives on time.

4.1.2 Deliverable

The main deliverable of this project is an open-source software package that achieves at-least the main goal of this project. That is, to successfully make use of side-information (hand-pose data only available during testing) to improve output performance of at-least one modality (i.e. hand-pose or action class). As an extension, the model should successfully show improvements in the output performance of both modalities (i.e. hand-pose and action class) with the use of side-information (i.e. hand-pose data during testing). As a further extension the project should explore the use of state-of-art and/or novel techniques to further extend performance from baseline scores. The package will be published online and thus there won't be a need to distribute hard copies. The software will be primarily written using the Python scripting language with the help of PyTorch, a popular deep-learning library for Python.

Bibliography

[1] "GoPro | The world's most versatile action cameras." [Online]. Available: https://gopro.com/

[2] "iON UK | Full details of the iON SnapCam Camera." [Online]. Available: https://uk.ioncamera.com/snapcam/

[3] "What is vlog (video blog)? - Definition from WhatIs.com." [Online]. Available: https://whatis.techtarget.com/definition/vlog-video-blog

[4] "Stereo depth technology - Intel RealSense Depth & Tracking Cameras." [Online]. Available: https://realsense.intel.com/stereo/

[5] "Microsoft HoloLens | The leader in mixed reality technology." [Online]. Available: https://www.microsoft.com/en-us/hololens

[6] T. Ishihara, K. M. Kitani, W. C. Ma, H. Takagi, and C. Asahawa, "Recognizing hand-object interactions in wearable camera videos," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2015-Decem, 2015, pp. 1349–1353. [Online]. Available: http://www.cs.cmu.edu/{~}kkitani/pdf/IKMTA-ICIP2015.pdf

[7] M. Cai, K. M. Kitani, and Y. Sato, "Understanding Hand-Object Manipulation with Grasp Types and Object Attributes," in *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016. [Online]. Available: http://www.roboticsproceedings.org/rss12/p34.pdf

[8] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," may 2015. [Online]. Available: http://arxiv.org/abs/1506.00019

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," dec 2013. [Online]. Available: https://arxiv.org/abs/1312.6114http://arxiv.org/abs/1312.6114

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," jun 2014. [Online]. Available: http://arxiv.org/abs/1406.2661

[11] Y. Gal and Z. A. Uk, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning Zoubin Ghahramani," Tech. Rep., 2016. [Online]. Available: http://yarin.co.

[12] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Networks*, vol. 22, no. 5-6, pp. 544–557, jul 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608009001130

[13] V. Vapnik, R. Izmailov, A. Gammerman, and V. Vovk, "Learning Using Privileged Information: Similarity Control and Knowledge Transfer," Tech. Rep., 2015. [Online]. Available: http://www.jmlr.org/papers/volume16/vapnik15b/vapnik15b.pdf

[14] D. Wu and L. Shao, "Leveraging Hierarchical Parametric Networks for Skeletal Joints Based Action Segmentation and Recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014, pp. 724–731. [Online]. Available: http://ieeexplore.ieee.org/document/6909493/

[15] Z. Shi and T. K. Kim, "Learning and refining of privileged information-based RNNs for action recognition from depth sequences," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 4684–4693. [Online]. Available: http://openaccess.thecvf.com/content{_}cvpr{_}2017/papers/Shi{_}Learning{_}and{_}Refining{_}CVPR{_}2017{_}paper.pdf

[16] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," apr 2017. [Online]. Available: https://arxiv.org/abs/1704.02463

[17] C. Zimmermann and T. Brox, "Learning to Estimate 3D Hand Pose from Single RGB Images," may 2017. [Online]. Available: http://arxiv.org/abs/1705.01389

[18] M. Oberweger and V. Lepetit, "DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, vol. 2018-Janua. IEEE, oct 2017, pp. 585–594. [Online]. Available: http://ieeexplore.ieee.org/document/8265285/

[19] G. Rogez, J. S. Supancic, M. Khademi, J. M. M. Montiel, and D. Ramanan, "3D Hand Pose Detection in Egocentric RGB-D Images," nov 2014. [Online]. Available: http://arxiv.org/abs/1412.0065

[20] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8121994/

[21] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing Nets: Combining GANs and VAEs with a Shared Latent Space for Hand Pose Estimation," feb 2017. [Online]. Available: http://arxiv.org/abs/1702.03431

[22] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T.-K. Kim, "Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals," dec 2017. [Online]. Available: http://arxiv.org/abs/1712.03917

[23] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Procedings of the British Machine Vision Conference 2011*. British Machine Vision Association, 2011, pp. 101.1–101.11. [Online]. Available: http://www.bmva.org/bmvc/2011/proceedings/paper101/index.html

[24] C. Keskin, F. Kıraç, Y. E. Kara, L. Akarun, Y. E. Kara, and L. Akarun, "Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests," in *Proceedings of the 12th European conference on Computer Vision - Volume Part VI*. Springer-Verlag, 2012, pp. 852–863. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33783-3{_}61

[25] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-Time Articulated Hand Pose Estimation Using Semi-supervised Transductive Regression Forests," in *2013 IEEE International Conference on Computer Vision*. IEEE, dec 2013, pp. 3224–3231. [Online]. Available: http://ieeexplore.ieee.org/document/6751512/

[26] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014, pp. 3786–3793. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909879

[27] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015, pp. 3325–3333. [Online]. Available: http://ieeexplore.ieee.org/document/7410737/

[28] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 4150–4158. [Online]. Available: http://ieeexplore.ieee.org/document/7780819/

[29] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," feb 2015. [Online]. Available: http://arxiv.org/abs/1502.06807

[30] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang, "Hand3D: Hand Pose Estimation using 3D Neural Network," apr 2017. [Online]. Available: http://arxiv.org/abs/1704.02224

[31] G. Moon, J. Y. Chang, and K. M. Lee, "V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map," nov 2017. [Online]. Available: http://arxiv.org/abs/1711.07399

[32] Y. Wu, W. Ji, X. Li, G. Wang, J. Yin, and F. Wu, "Context-Aware Deep Spatio-Temporal Network for Hand Pose Estimation from Depth Images," oct 2018. [Online]. Available: https://arxiv.org/abs/1810.02994

[33] Y. Zhou, J. Lu, K. Du, X. Lin, Y. Sun, and X. Ma, "HBE: Hand Branch Ensemble Network for Real-Time 3D Hand Pose Estimation," in *Computer Vision – ECCV 2018*. Springer, Cham, sep 2018, pp. 521–536. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01264-9{_}31

[34] S. Baek, K. I. Kim, and T.-K. Kim, "Augmented Skeleton Space Transfer for Depth-based Hand Pose Estimation," may 2018. [Online]. Available: http://arxiv.org/abs/1805.04497

[35] X. Chen, "Awesome Work on Hand Pose Estimation." [Online]. Available: https://xinghaochen.github.io/awesome-hand-pose-estimation/

[36] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose Guided Structured Region Ensemble Network for Cascaded Hand Pose Estimation," aug 2017. [Online]. Available: http://arxiv.org/abs/1708.03416

[37] H. Guo, G. Wang, X. Chen, and C. Zhang, "Towards Good Practices for Deep 3D Hand Pose Estimation," jul 2017. [Online]. Available: http://arxiv.org/abs/1707.07248

[38] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based Deep Hand Pose Estimation," jun 2016. [Online]. Available: http://arxiv.org/abs/1606.06854

[39] M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Transactions on Computers*, vol. C-22, no. 1, pp. 67–92, jan 1973. [Online]. Available: http://ieeexplore.ieee.org/document/1672195/

[40] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D Hand Pose Estimation using Point Sets," in *CVPR*, 2018, pp. 8417–8426. [Online]. Available: http://openaccess.thecvf.com/content{_}cvpr{_}2018/papers/Ge{_}Hand{_}PointNet{_}3D{_}CVPR{_}2018{_}paper.pdf

[41] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," dec 2016. [Online]. Available: https://arxiv.org/pdf/1612.00593.pdfhttp://arxiv.org/abs/1612.00593

[42] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-Based Hand Pose Estimation: Data, Methods, and Challenges," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015, pp. 1868–1876. [Online]. Available: https://www.cv-foundation.org/openaccess/content{_}iccv{_}2015/papers/Supancic{_}Depth-Based{_}Hand{_}Pose{_}ICCV{_}2015{_}paper.pdfhttp://ieeexplore.ieee.org/document/7410574/

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/https://arxiv.org/abs/1512.03385

[44] Ying Wu, J. Lin, and T. Huang, "Capturing natural hand articulation," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2. IEEE Comput. Soc, 2001, pp. 426–432. [Online]. Available: http://ieeexplore.ieee.org/document/937656/

[45] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, no. 1, pp. 53–58, jan 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900142

[46] C. Xu and L. Cheng, "Efficient Hand Pose Estimation from a Single Depth Image," in *2013 IEEE International Conference on Computer Vision*. IEEE, dec 2013, pp. 3456–3462. [Online]. Available: http://ieeexplore.ieee.org/document/6751541/

[47] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 1–10, 2014. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/tompson-siggraph-14.pdfhttp://dl.acm.org/citation.cfm?doid=2672594.2629500

[48] X. Sun, Y. Wei, Shuang Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015, pp. 824–832. [Online]. Available: http://ieeexplore.ieee.org/document/7298683/

[49] L. G. Shapiro and G. C. Stockman, "Computer vision," p. 580, 2001. [Online]. Available: https://dl.acm.org/citation.cfm?id=558008http://nana.lecturer.pens.ac.id/index{_}files/referensi/computer{_}vision/ComputerVision.pdf

[50] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, apr 1985. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/0734189X85900167

[51] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, jan 2005. [Online]. Available: http://link.springer.com/10.1023/B:VISI.0000042934.15159.49

[52] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," jun 2014. [Online]. Available: http://arxiv.org/abs/1406.2199

[53] R. Ghosh, "Deep Learning for Videos: A 2018 Guide to Action Recognition." [Online]. Available: http://blog.qure.ai/notes/deep-learning-for-videos-action-recognition-review

[54] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015, pp. 4489–4497. [Online]. Available: https://arxiv.org/pdf/1412.0767.pdf

[55] Y. Kong and Y. Fu, "Human Action Recognition and Prediction: A Survey," jun 2018. [Online]. Available: https://arxiv.org/pdf/1806.11230.pdfhttp://arxiv.org/abs/1806.11230

[56] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, may 2017, pp. 4724–4733. [Online]. Available: https://arxiv.org/pdf/1705.07750.pdfhttp://arxiv.org/abs/1705.07750

[57] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 2625–2634. [Online]. Available: http://jeffdonahue.com/lrcn/.

[58] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks," mar 2016. [Online]. Available: www.aaai.orghttp://arxiv.org/abs/1603.07772

[59] B. Ghojogh and H. Mohammadzade, "Automatic extraction of key-poses and key-joints for action recognition using 3D skeleton data," in *2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP)*. IEEE, nov 2017, pp. 164–170. [Online]. Available: https://ieeexplore.ieee.org/document/8342342/

[60] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "Lie-X: Depth Image Based Articulated Object Pose Estimation, Tracking, and Action Recognition on Lie Groups," *International Journal of Computer Vision*, vol. 123, no. 3, pp. 454–478, jul 2017. [Online]. Available: http://link.springer.com/10.1007/s11263-017-0998-6

[61] Y. Tang, Z. Wang, J. Lu, J. Feng, and J. Zhou, "Multi-stream Deep Neural Networks for RGB-D Egocentric Action Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8489917/

[62] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-Based Action Recognition with Spatial Reasoning and Temporal Stack Learning," Tech. Rep., 2018. [Online]. Available: https://arxiv.org/pdf/1805.02335.pdf

[63] C. Xie, C. Li, B. Zhang, C. Chen, J. Han, C. Zou, and J. Liu, "Memory Attention Networks for Skeleton-based Action Recognition," apr 2018. [Online]. Available: http://arxiv.org/abs/1804.08254

[64] M. Liu and J. Yuan, "Recognizing Human Actions as the Evolution of Pose Estimation Maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 18)*. IEEE, jun 2018, pp. 1159–1168. [Online]. Available: https://ieeexplore.ieee.org/document/8578225/

[65] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose MoTion Representation for Action Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018, pp. 7024–7033. [Online]. Available: https://ieeexplore.ieee.org/document/8578832/

[66] Denny Britz, "Attention and Memory in Deep Learning and NLP – WildML." [Online]. Available: http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/

[67] Adam Kosiorek, "Attention in Neural Networks and How to Use It." [Online]. Available: http://akosiorek.github.io/ml/2017/10/14/visual-attention.html

[68] W. Du, Y. Wang, and Y. Qiao, "RPAN: An End-to-End Recurrent Pose-Attention Network for Action Recognition in Videos," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, 2017, pp. 3745–3754. [Online]. Available: http://openaccess.thecvf.com/content{_}ICCV{_}2017/papers/Du{_}RPAN{_}An{_}End-To-End{_}ICCV{_}2017{_}paper.pdf

[69] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned Spatio-Temporal Attention for Human Action Recognition," mar 2017. [Online]. Available: http://arxiv.org/abs/1703.10106

[70] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," sep 2014. [Online]. Available: https://arxiv.org/pdf/1409.0473.pdfhttp://arxiv.org/abs/1409.0473

[71] Cagatay Odabasi, "Skeleton-Based-Action-Recognition-Papers-and-Notes." [Online]. Available: https://blog.coast.ai/five-video-classification-methods-implemented-in-keras-and-tensorflow-99cad29cc0b5

[72] Duohan Liang, "Skeleton-Based-Action-Recognition-Papers." [Online]. Available: https://github.com/XiaoCode-er/Skeleton-Based-Action-Recognition-Papers

[73] C. Cao, Y. Zhang, Y. Wu, H. Lu, and J. Cheng, "Egocentric Gesture Recognition Using Recurrent 3D Convolutional Neural Networks with Spatiotemporal Transformer Modules," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, 2017, pp. 3783–3791. [Online]. Available: http://openaccess.thecvf.com/content{_}ICCV{_}2017/papers/Cao{_}Egocentric{_}Gesture{_}Recognition{_}ICCV{_}2017{_}paper.pdf

[74] T. Chalasani, J. Ondrej, and A. Smolic, "Egocentric Gesture Recognition for Head-Mounted AR devices," 2018. [Online]. Available: https://arxiv.org/pdf/1808.05380.pdfhttp://arxiv.org/abs/1808.05380

[75] S. Sudhakaran and O. Lanz, "Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition," jul 2018. [Online]. Available: http://arxiv.org/abs/1807.11794

[76] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, jul 2018, pp. 3–19. [Online]. Available: http://arxiv.org/abs/1807.06521

[77] U. Iqbal, M. Garbade, and J. Gall, "Pose for Action - Action for Pose," mar 2016. [Online]. Available: http://arxiv.org/abs/1603.04037

[78] Y. Yang, "Articulated pose estimation with flexible mixtures-of-parts resenting shape," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011. [Online]. Available: https://vision.ics.uci.edu/papers/YangR{_}CVPR{_}2011/YangR{_}CVPR{_}2011.pdfhttp://www.mendeley.com/research/articulated-pose-estimation-flexible-mixturesofparts-resenting-shape/

[79] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning," Tech. Rep. [Online]. Available: http://openaccess.thecvf.com/content{_}cvpr{_}2018/papers/Luvizon{_}2D3D{_}Pose{_}Estimation{_}CVPR{_}2018{_}paper.pdf

[80] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-Modal Deep Variational Hand Pose Estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018, pp. 89–98. [Online]. Available: https://arxiv.org/pdf/1803.11404.pdfhttp://arxiv.org/abs/1803.11404https://ieeexplore.ieee.org/document/8578115/

[81] M. Abdi, E. Abbasnejad, C. P. Lim, and S. Nahavandi, "3D Hand Pose Estimation using Simulation and Partial-Supervision with a Shared Latent Space," jul 2018. [Online]. Available: http://arxiv.org/abs/1807.05380

[82] C. Doersch, "Tutorial on Variational Autoencoders," jun 2016. [Online]. Available: http://arxiv.org/abs/1606.05908

[83] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognition*, vol. 77, pp. 329–353, may 2018. [Online]. Available: http://arxiv.org/abs/1612.03365http://dx.doi.org/10.1016/j.patcog.2017.10.009https://linkinghub.elsevier.com/retrieve/pii/S0031320317304065