

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Dipartimento di Informatica - Scienza e Ingegneria - DISI

Corso di Laurea Magistrale in Informatica

**STUDY AND ANALYSIS OF HEAD POSE
ESTIMATION METHODS AND DATABASES**

Relatore:

Prof. ANDREA ASPERTI

Presentata da:

DANIELE FILIPPINI

Sessione III

Anno Accademico 2020/2021

Sommario

Il campo della stima dell'orientamento del volto, o Head Pose Estimation (HPE), é una popolare ed attiva area di ricerca. Durante gli anni molti approcci sono stati costantemente sviluppati, portando ad un progressivo aumento dell'accuratezza delle predizioni; nonostante ciò la stima dell'orientamento del volto rimane un argomento di ricerca aperto, soprattutto in ambienti non vincolati. In questa tesi, esamineremo la crescente quantità di dataset disponibili e le metodologie utilizzate per l'acquisizione delle annotazioni (etichette) relative all'orientamento del volto. Discuteremo l'evoluzione del campo proponendo una classificazione dei metodi per Head Pose Estimation e spiegandone i relativi vantaggi e svantaggi, il tutto con un focus principale sulle recenti tecniche basate sul deep learning. Alla fine del lavoro viene poi presentato un confronto e una discussione approfonditi delle prestazioni. La tesi indica anche direzioni promettenti per la ricerca futura sull'argomento.

Abstract

Head pose estimation is an active and popular area of research. Over the years many approaches have constantly been developed, leading to a progressive improvement in accuracy; nevertheless, head pose estimation remains an open research topic, especially in unconstrained environments. In this thesis, we will review the increasing amount of available datasets and the methodologies used to acquire ground-truth annotations. We will discuss the evolution of the field by proposing a classification of head pose estimation methods and by explaining their advantages and disadvantages, all with a main focus on the recent deep learning based techniques. An in-depth performance comparison and discussion is presented at the end of the work. The thesis also states promising directions for future research on the topic.

Key words: Head pose estimation, Head pose database, Face analysis, Face alignment, Face landmark detection, Deep learning, Convolutional neural networks

Contents

1	Introduction	1
1.1	Motivation	2
2	Background on Machine Learning	7
2.1	Neural Networks	8
2.2	Convolutional Neural Networks	10
2.2.1	Fully Connected Layers	11
2.2.2	Convolutional Layers	12
2.2.3	Pooling Layers	15
2.2.4	Non-Linear Activation Layers	16
2.2.5	Normalization Layers	17
2.3	Data Augmentation	18
2.4	Transfer Learning	19
2.5	Multi-Task Learning	21
3	Background on Face Vision	25
3.1	Face Related Computer Vision Tasks	25
3.1.1	Facial Analysis	29
3.1.2	Data Augmentation	31
3.2	3D Morphable Face Models	33
3.3	3D Dense Face Alignment	35
4	Head Pose Estimation	37
4.1	Datasets	40

4.2	Head pose rotations representations	53
4.2.1	Euler angles	53
4.2.2	Rotation matrix	55
4.2.3	Quaternions	56
4.3	Creating Ground-Truth data	57
4.4	Evaluation metrics	65
4.5	Methods	67
4.6	Evaluation pipelines	95
4.7	Discussion	99
5	Conclusion	111
	Bibliography	113
A	Links to datasets	139

List of Figures

1.1	Example of driving application	3
1.2	Example of tasks strongly linked to head pose estimation	4
2.1	Shallow and Deep Networks	9
2.2	Neural Networks Types	10
2.3	CNNs base model	11
2.4	Example of Dense Layer	12
2.5	Example of Convolutional Layer	13
2.6	Example of Convolutional Filter	14
2.7	Dense vs Sparse connectivity	15
2.8	Example of Pooling	16
2.9	Examples of activation functions	17
2.10	Data augmentation example	18
2.11	Visualization of Convolutional Kernels at different levels in a CNN .	20
2.12	Transfer Learning methods for CNNs	21
2.13	Differences between MTL and other learning paradigms	22
3.1	Example of different Computer Vision face related tasks	28
3.2	Example of face variations	30
3.3	An overview of transformation types	33
3.4	Evolution of 3D Morphable Models	34
3.5	Example of 3D dense face alignment	36
4.1	Example of coarse and fine head pose estimation	38

4.2	Euler angles in Head Pose Estimation	38
4.3	Euler angles different rotations	54
4.4	Euler angles	55
4.5	Example of manual labelling process	58
4.6	Synthetic Head Pose dataset generation	59
4.7	Example of errors in Pointing'04 head pose database	60
4.8	Some methods for Head Pose datasets creation	61
4.9	Other methods for Head Pose datasets creation	63
4.10	Pose labelling using ICP algorithm	64
4.11	Cost comparison of annotations acquisition	65
4.12	Appearance template method	70
4.13	Detector array method	71
4.14	Manifold embedding method	74
4.15	Tracking method	76
4.16	Segmentation based method	78
4.17	Geometrical method	79
4.18	Model based methods	84
4.19	Non-linear regression method: Hopenet	86
4.20	Non-linear regression method: POSEidon	88
4.21	Multi-task methods	92
4.22	Example of inaccuracies in ground-truth annotations	100
4.23	Influence of bbox margin and background on Head Pose Estimation	106
4.24	Comparison of pose estimation results with MAE and MAEV evaluation metrics	108

List of Tables

4.1	Available datasets for Head Pose Estimation	43
4.2	Head Pose Estimation publications	95
4.3	Evaluation results of Head Pose Estimation on AFLW2000 and BIWI	97
4.4	Evaluation results of Head Pose Estimation on AFLW	99
4.5	Evaluation results of Head Pose Estimation on other databases . . .	103
4.6	Comparisons of different landmark detectors	107
A.1	Links to available datasets for Head Pose Estimation	140

Chapter 1

Introduction

The capacity to estimate the head pose of another person is a common human ability that presents a unique challenge for computer vision systems. People have the ability to quickly and effortlessly interpret the orientation and movement of a human head, thereby allowing one to infer the intentions of others who are nearby and to comprehend an important non-verbal form of communication.

Head pose is an important cue in computer vision when using facial information and has a wide variety of uses in human-computer interaction.

Over the last three decades, methods for head pose estimation have received increasing attention due to their application in several image analysis tasks. Although many techniques have been developed in the years to address this issue, head pose estimation remains an open research topic, particularly in unconstrained environments (due to internal and external conditions and complex imaging feature face [57]).

In a computer vision context, *head pose estimation* (HPE) is the process of inferring the orientation of a human head from digital imagery. Like other facial vision tasks, an ideal head pose estimator must demonstrate invariance to a variety of image-changing factors, such as camera distortion, projective geometry, multi-source non-Lambertian lighting, as well as biological appearance, facial expression, and the presence of accessories like glasses and hats [55].

1.1 Motivation

HPE systems play an important role in the development of different intelligent environments, so that several computer vision applications rely on a robust HPE system as a prerequisite: for example, applications of gaze estimation [91], virtual/augmented reality [95], and human-computer interaction [94], strongly benefit from knowing the exact position of the head in 3D space.

Some application examples are:

- **Human Social Behaviour Analysis:** People use the orientation of their heads to convey rich, inter-personal information. For example, there is important meaning in the movement of the head as a form of gesturing in a conversation [92] to indicate when to switch roles and begin speaking or to indicate who is the intended target subject. People nod to indicate that they understand what is being said, and they use additional gestures to indicate dissent, confusion, consideration, and agreement [93].

In addition to the information that is implied by deliberate head gestures, there is much that can be inferred by observing a person's head. For instance, quick head movements may be a sign of surprise or alarm, these could also trigger reflexive responses from other observers [90].

Therefore, HPE can be used in smart rooms to monitor participants in a meeting and to record their activities, in particular, their attention can be indirectly related to their head pose [87]. Systems exploiting head pose estimation to analyse people's behaviour and human interaction in meetings and workplaces have been proposed in [88] [89] [145] [145].

There are also studies on systems for automatic pain monitoring that show how including head pose can improve the performance for both person-specific and general classifiers [36].

- **Driving Safety & Assistance:** HPE systems are particularly useful for assisting drivers by providing contextual alert signals, for example in the case of pedestrians outside the driver's field of view [48].

Moreover, the head pose can give clues about the intention of the pedestrian e.g. a pedestrian will wait for a stopped automobile driver to look at him before stepping into a crosswalk (this is an example of pattern recognition), very important also in the case of autonomous vehicles.

Applications to infer the driver's pose are very important for safety, as they can provide insights about distraction, intention, sleepiness, awareness or detect blind spots of the driver [146], for this reason, in recent years many datasets that address this specific scenario have been published [32] [33] [35].

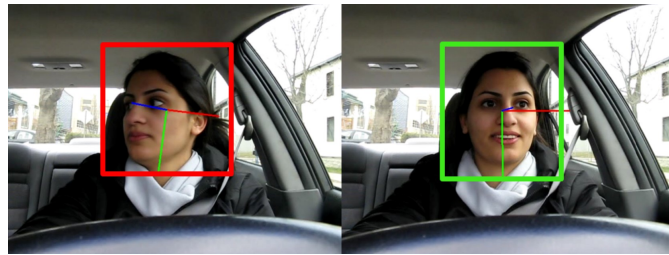


Figure 1.1: An example of application to driver assistance. Right: Green box indicates yaw $< \pm 45^\circ$ and potential awareness of vehicle. Left: Red box indicates possible inattention (image from [61]).

- **Surveillance and Safety:** Head pose estimation in surveillance video images is an important task in computer vision because it tracks the visual attention and provides insight on human behavioural intentions [83] [84]. Systems for direct an automated surveillance network have been proposed in [85] [86].
- **Targeted Advertisement:** Methods to track visual attention in wandering people have been proposed in the literature [82]. These systems count people looking at particular outdoor advertisements (targeted advertisement) and can determine what a person is looking at if movement is unconstrained. Systems like these can be used for behaviour analysis and cognitive science in real-world applications also in indoor environments, such as TV viewers' behaviour analysis [81].

- **Interface Design:** By perceiving the human attention when they look at an interface (e.g. the page of web or software), it is possible to evaluate the property and significance of the displayed visual elements and further guide the design or rearrangement of these elements [80].

Therefore, head pose estimation can get used to monitor human social activities, to observe the behaviour of specific targets, but also to enhance the function of some face-related tasks, including expression detection, gaze estimation, full-body pose estimation (as shown in figure 1.2) and identity recognition.

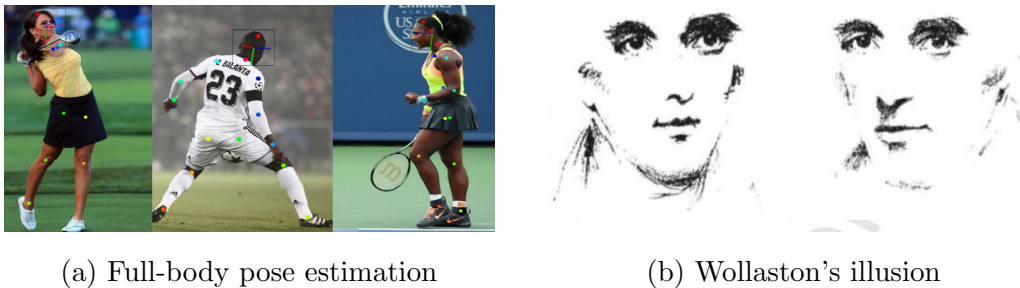


Figure 1.2: Example of tasks strongly linked to head pose estimation: (a) Frequently human pose estimators do not estimate sufficient keypoints for accurate HPE, for this reason integrating specific methods could be beneficial, for example, in sports broadcasting or by coaching staff to estimate participants fields of views and situational awareness when analysing plays [61]; (b) Despite the eyes are in the same position in both face images, the perception is that the two gazes are differently oriented. Gaze prediction comes from a combination of both eyes and head pose direction [58].

The intrinsic interaction between head pose and other face parts is also confirmed in more recent research. Studies in [96] [97] [98] [99] suggest that the mutual relationship between face parts can be exploited not only for HPE, but also for other visual tasks such as gender recognition, race classification, and age estimation, making head pose estimation a very important and useful task.

The contributions of the thesis are:

- a complete and updated review of all the available databases for the head pose estimation task with an exhaustive explanation of different acquisition methods;
- a categorization and explanation of the different approaches used in the literature for head pose estimation, with a specific focus on modern deep learning approaches;
- report and discussion of most existing head pose estimation methods and their performance on common datasets;

The remainder of the thesis is organized as follows: *Chapter 2* contains an introduction to some common concepts of the machine learning field; *Chapter 3* discusses some preliminary concepts of the computer vision field related to facial analysis; in *Chapter 4* existing datasets, acquisition methods, recent and prominent approaches for head pose estimation are reported and discussed; *Chapter 5* concludes the thesis summarizing the contribution of the proposed work and highlighting potential future directions to explore. *Appendix A* contains a table with currently available links to download the presented datasets.

Note: All numerical results reported in the following tables are borrowed from the original publications.

Chapter 2

Background on Machine Learning

The last few years have seen an increasing interest of the artificial intelligence community for deep learning techniques. These computational models are representation-learning methods with multiple levels of representation, obtained by composing multiple non-linear processing layers that can learn hierarchical representations with increasing levels of abstraction. The key aspect of deep learning is that, differently from conventional feature, these layers of features are not designed by humans, instead they are learned directly from data. Convolutional neural networks are deep feed-forward neural network architectures that are easy to train and generalize much better than common neural networks. These architectures have proved to be very effective in many tasks and they are widely adopted by the computer vision community.

It is assumed that the reader of this thesis has prior knowledge in the field of machine learning with a focus on deep learning. The apprehension is assumed to be at least on a level corresponding to a master's degree in computer science. Due to this assumption, basic concepts like, but not limited to, loss functions will be omitted or described on a very high level.

2.1 Neural Networks

Neural networks, also known as artificial neural networks (ANNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Neural networks are *weighted graphs*. They consist of an ordered set of *layers*, where every layer is a set of *nodes*. The first layer of the neural network is called the *input layer*, the last one is called the *output layer*, and the layers in between are called *hidden layers*. Layers are semantic groups of nodes. Nodes belonging to one layer are connected to the nodes in the following and/or the previous layers. These connections are *weighted edges*, and they are referred to as *weights*.

Given an input, neural network nodes return outputs, which are real numbers. The output of a node is calculated by applying a function α (called *activation function*) to the *weighted sum* of outputs of the nodes belonging to the previous layer. Preceding that, the output of the input layer ($o(0)$) is equal to the input. By calculating the layers outputs consecutively we calculate the result returned by the output layer. This process is called *inference*. Therefore a Neural Network is just a mathematical function mapping some set of input values to output values. The goal is to approximate some function f^* . These are called networks because they can be represented by composing together many functions. Indeed, we can see each layer as a function and these functions are connected in a chain to form $f(x) = f^n(f^{n-1}(\dots f^1(x)\dots))$, where f^1 is the first layer of the network, f^2 is the second, and so on [63].

For example, for a classifier, $y = f^*(x)$ maps an input x to a category y . A neural network defines a mapping $y = f(x, \theta)$ and learns the value of the parameters θ that result in the best function approximation.

Neural networks can be more complex and this complexity is added by the addition of more hidden layers. A neural network that is made up of more than three layers i.e. has one input layer, several hidden layers, and one output layer is known as a *Deep Neural Network*. These networks are what support and underpin the idea and concepts of Deep Learning where the model basically trains itself to

process and predict from data.

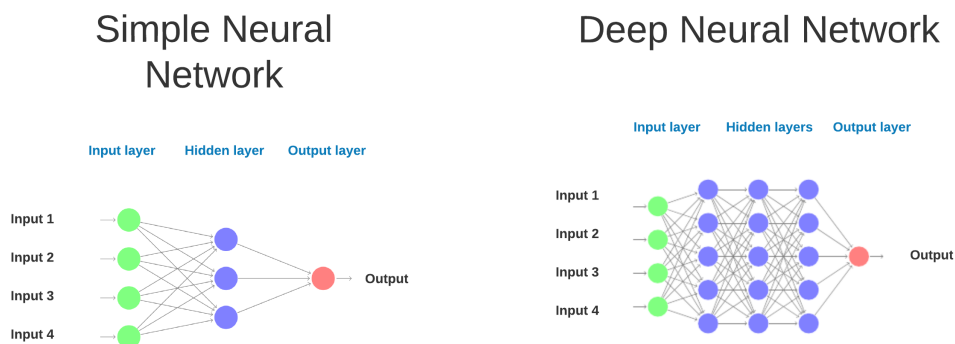


Figure 2.1: An example of Shallow network and Deep network

There are different types of Neural Networks, the main three types that have demonstrated impressive performance in complex machine learning tasks, such as image classification or speech recognition, are:

- **Feed-Forward NN:** also called Multi-Layer Perceptron (MLP), is the most basic deep neural network, it's composed of *fully connected* layers and the input goes from the left to the right, there are no backward connections (the network is acyclic).
- **Convolutional NN:** is the most commonly employed type of deep neural network in Computer Vision tasks, but can also be used for other types of input like audio. This is a feed-forward neural network that has one or multiple *convolutional layers*. Using this kind of layers the network is able to capture the high-level representation of the input data, making it able to solve complex tasks, such as image classification, object detection, face authentication, etc.
- **Recurrent NN:** is another class of artificial neural networks designed to recognize patterns in sequences of data e.g. in text, handwriting, spoken words, etc. In this kind of network, there are *backward connections* (the network is cyclic), the input of a RNN consists of the current input and the

previous samples. Each neuron owns an internal memory (*hidden state*) that keeps the information of the computation from the previous samples.

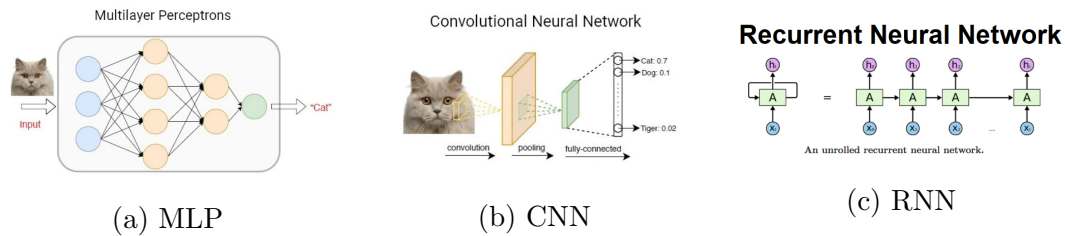


Figure 2.2: Three different types of neural network

2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a specialized kind of feed-forward neural network for processing data that has a known grid-like topology, such as images, videos and time-series. The name “convolutional neural network” indicates that the network employs a specialized kind of linear operation called *convolution*. At the most basic level, a convolutional neural network is a multi-layer, hierarchical neural network. There are only three peculiarities that distinguish CNNs from simple feed-forward neural networks: sparse connectivity, weight sharing, and spatial pooling or sub-sampling layers.

A modern deep convolutional neural network consists of several layers, as shown in Fig. 2.3. Several stages of convolution, non-linearity are stacked, followed by more convolutional and fully-connected layers. Intuitively, the low-level convolutional filters, such as those in the first convolutional layers, can provide a low-level encoding of the input data, mid-level filters compose the previous information to a higher level of abstraction and by moving to higher layers more and more complicated structures are encoded. In the case of image data, local combinations of edges forms motifs, motifs assemble into parts and parts compose objects. In addition to convolutional and fully-connected layers, various optional layers can be considered such as *pooling* and *normalization* layers. The following sections describe in detail components characterizing a classic CNN.

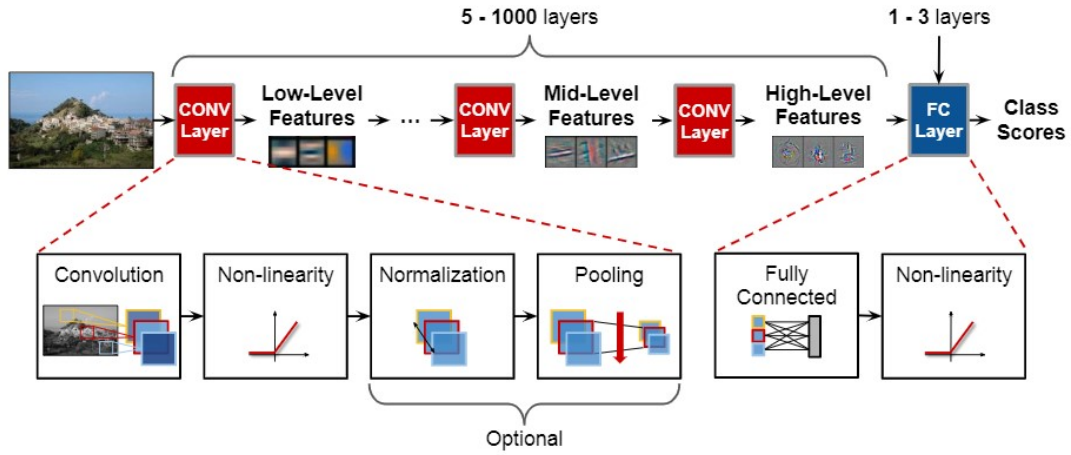


Figure 2.3: CNNs base model

2.2.1 Fully Connected Layers

As the name suggests, for two consecutive layers to be *fully connected*, all the nodes in layer $l^{(k)}$ must be connected to all the nodes in the following layer $l^{(k+1)}$.

The weight matrix connecting these Fully Connected layers, or Dense Layers, is defined as $w^{(k)} \in \mathbb{R}^{m^{(k-1)} \times m^{(k)}}$.

Most fully connected layers also include a bias term ($b^{(k)} \in \mathbb{R}^{m^{(k)}}$) to account for the constants in the system.

The output of a fully connected layer, $o^{(k)}$ is

$$o^{(k)} = (o^{(k-1)})^T w^{(k)} + b^{(k)}$$

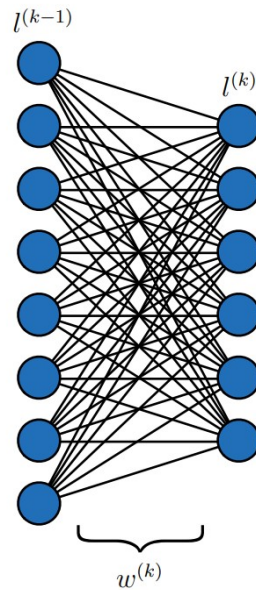


Figure 2.4: An example of two Fully Connected layers, $l^{(k-1)}$ and $l^{(k)}$, connected by the weight matrix $w^{(k)}$

2.2.2 Convolutional Layers

Convolutional Layer is the building block of a Convolutional Neural Network. In the context of a convolutional neural network, a *convolution* is the simple application of a *filter* to an input that results in an *activation*. Repeated application of the same filter to an input results in a map of activations called a *feature map*, indicating the locations and strength of a detected feature in an input, such as an image.

The technique was designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a *filter* or a *convolutional kernel*.

The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a *dot product*. A dot product is an element-wise multiplication between the filter-sized patch of the input and filter, which is then summed, always resulting in a single value. Because

it results in a single value, the operation is often referred to as the “scalar product”.

Using a filter smaller than the input is intentional as it allows the same filter (set of weights) to be multiplied by the input array multiple times at different points on the input. Specifically, the filter is applied systematically to each overlapping part or filter-sized patch of the input data, left to right, top to bottom.

This systematic application of the same filter across an image is a powerful idea. If the filter is designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image.

Indeed, pixels that are close together in an image (e.g. adjacent pixels) tend to be strongly correlated and can represent meaningful features such as edges, while pixels that are far apart in the image tend to be weakly correlated or uncorrelated.

Therefore, each neuron at layer $l^{(k)}$ is connected via a parametric *kernel* to a fixed subset of neurons at layer $l^{(k-1)}$, this subset is called *receptive field*. The kernel is convolved over the whole previous layer.

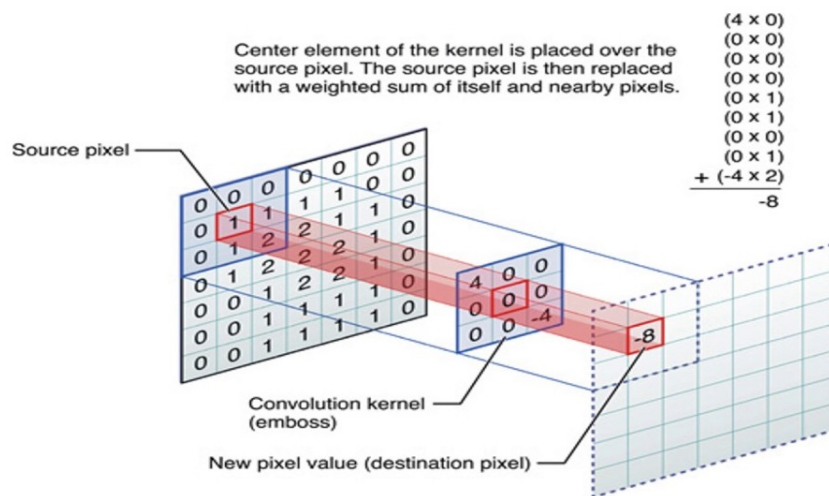


Figure 2.5: An example of a Convolutional Layer

Convolutional Neural Networks do not learn a single filter, in fact, they learn multiple features in parallel for a given input. A Convolutional Layer simultane-

ously applies multiple trainable filters to its inputs, this diversity allows specialization (detecting multiple features), e.g. not just lines, but the specific lines seen in specific training data.

Colour images are composed of multiple sub-layers, typically one for each *colour channel*, such as red, green, and blue (RGB). Grayscale images have just one channel, but some images may have much more.

From a data perspective, this means that a single image provided as input to the model is, in fact, three images.

A filter must always have the same number of channels as the input, often referred to as *filter depth*.

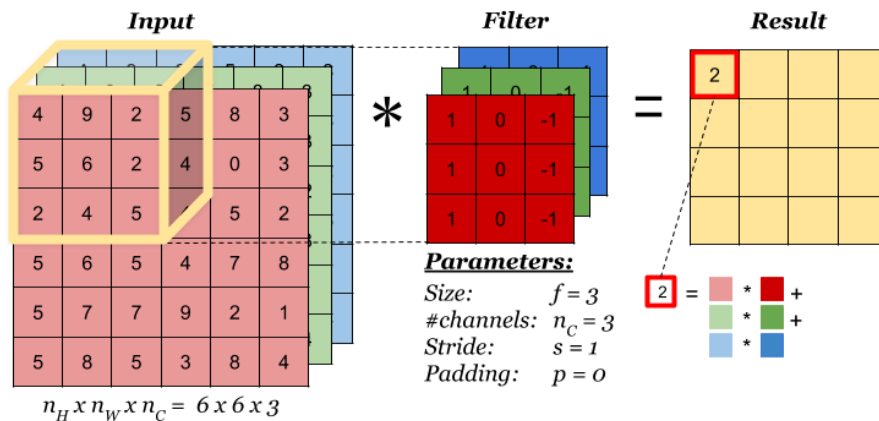


Figure 2.6: An example of a Convolutional Filter

The two main advantages of a Convolutional Layer over a Fully Connected Layer are:

- **Parameters sharing:** maintain the same feature detector used in one part of the input data across other sections of the input;
- **Sparsity of the connections:** each neuron is connected only with its receptive field;

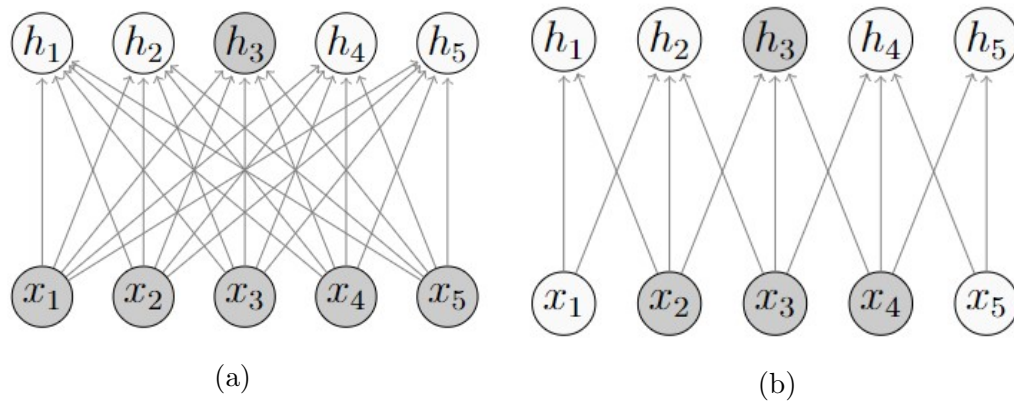


Figure 2.7: Dense vs Sparse connectivity. Input neurons in x that affect the output h_3 have been highlighted. In Dense connectivity (a) all the inputs affect h_3 . In Sparse connectivity (b) only three input neurons affect h_3 , as this is formed by convolution with a kernel of width 3.

The hyperparameters of Convolutional Layers are: *spatial filter size*, *depth*, *stride* and *padding*. Filter size corresponds to the spatial extent (width and height) of the filters that are convolved with the input image at different spatial locations. The depth of the output controls the number of filters that connect to the same region of the input volume. The stride controls the filter shift and determines the dimension of the resulting activation map, higher stride reduce receptive fields overlap and reduce spatial dimensions. The padding parameter allows to control the spatial size of activation maps by extending the input activation map. This is commonly done by adding zeros at activation map outer edges.

2.2.3 Pooling Layers

Pooling is a way of reducing the dimensionality of a layer, in order to reduce the computational load, the memory usage and the number of parameters (thereby limiting the risk of overfitting). Moreover, it also introduces some level of *invariance* to small translations in the input.

Just like in Convolutional Layers, each neuron in a Pooling Layer is connected to the outputs of a limited number of neurons in the previous layer, located within

a small rectangular receptive field. However, a pooling neuron has no weights, all it does is aggregate the inputs using an *aggregation function*, such as:

- **Max Pooling:** takes the maximum value in a channel within the receptive field;
- **Average Pooling:** averages the values within the receptive field per channel;

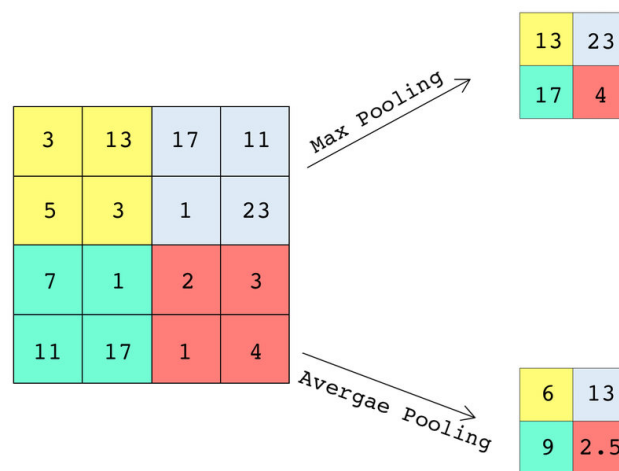


Figure 2.8: An example of the results for Max Pooling and Average Pooling

2.2.4 Non-Linear Activation Layers

A non-linear activation layer is usually applied after each convolutional layer or fully-connected layer. Various non-linear functions are used to introduce non-linearity into a CNN as shown in Figure 2.9. Traditional non-linear activation functions are *sigmoid* and *hyperbolic tangent*. These functions tend to saturate respectively at zero and one, and minus one and one, causing the so-called *vanishing gradient problem*: if the activity in the network during training is close to zero then the gradient for the sigmoid function may go to zero. For this reason, non-saturated activation functions such as the Rectified Linear Unit (ReLU) have been introduced [63]. ReLU is a piecewise linear function that prunes the negative

part to zero and retains the positive part. It allows a network to easily obtain sparse representation that is desirable because is more biologically plausible and leads to mathematical advantages, such as information disentangling and linear separability. Due to its simplicity and its stability to enable fast training, ReLU is the most used activation function at the moment.

Other variants of ReLU, such as LeakyReLU, PReLU or ExponentialReLU, are widely used and are some of the key-factors of surpassing human-level performance on some tasks [63].

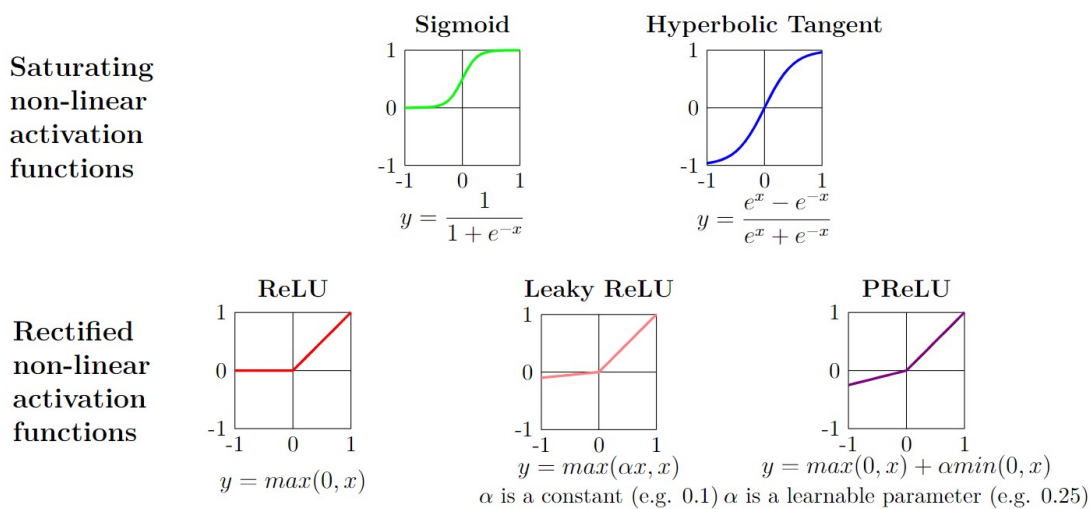


Figure 2.9: Various examples of Non-Linear Activation Functions

2.2.5 Normalization Layers

Normalization layer enables to control distribution across layers to significantly speed up training and improve performances. The distribution of input layers activations (σ, μ) is normalized such that it has zero-mean and a unit standard deviation.

In Batch Normalization (BN), now considered standard practice in the design of CNNs, the normalized value is further scaled and shifted. The parameter (γ, β) are learned during the training phase.

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}\gamma + \beta$$

Batch Normalization is usually performed between the convolutional or fully-connected layers and the non-linear function. It alleviates a lot of problems with properly initializing CNNs by explicitly forcing activations through a network to take on a unit normal distribution at the very beginning of training.

2.3 Data Augmentation

There exists a lot of ways to improve the results of a neural network by changing the way we train it. In particular, *data augmentation* is a common practice to virtually increase the size of the training dataset, since it is not always possible to get new data, or can be too expensive. Data augmentation is also used as a *regularization technique*, making the model more robust to slight changes in the input data.

In the data augmentation process some operations (rotation, zoom, shift, flips, etc.) are applied randomly to the input data. By this means, the model is never shown twice the exact same example and has to learn more general features about the classes it has to recognize.

Data augmentation is fundamentally important for improving the performance of neural networks in the following aspects:

1. It is inexpensive to generate a huge number of synthetic data with annotations in comparison to collecting and labelling real data.
2. Synthetic data can be accurate, so it has ground-truth by nature.

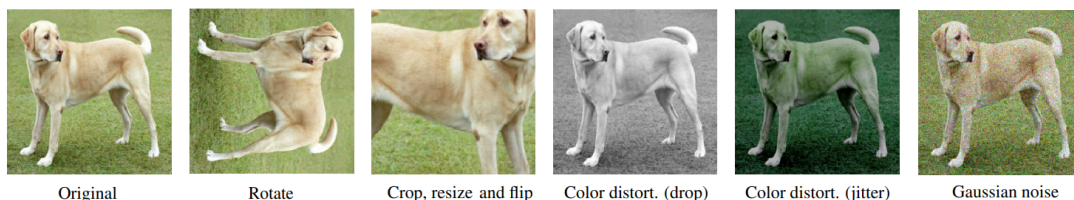


Figure 2.10: Data augmentation example

2.4 Transfer Learning

Training a convolutional neural network requires a huge amount of data, making computation particularly time-consuming. To overcome this problem the *transfer learning* technique is introduced. It consists of using knowledge from a similar task to solve a problem at hand. In practice, it usually means employing, as initializations for the deep neural network, the weights learned from a similar task, rather than starting from a random initialization of the weights, and then further training the model on the available labelled data to solve the task at hand.

Transfer learning enables to train models on datasets as small as a few thousand examples, and it can deliver a very good performance. Transfer learning from pre-trained models can be performed in three ways:

1. **Feature Extractors:** usually, the last layers of the neural network are doing the most abstract and task-specific calculations, which are generally not easily transferable to other tasks. By contrast, the initial layers of the network learn some basic features like edges and common shapes, which are easily transferable across tasks. We can see from figure 2.11 a hierarchical representation of this.

A common practice is to take a model pre-trained on large labelled image datasets (such as ImageNet [68]) and chop off the fully connected layers at the end. New, fully connected layers are then attached and configured according to the required number of classes. Transferred layers are frozen, and the new layers are trained on the available labelled data for the current task.

In this setup, the pre-trained model is being used as a feature extractor and the fully connected layers on the top can be considered as a shallow classifier. This set-up is more robust to overfitting as the number of trainable parameters is relatively small, so this configuration works well when the available labelled data is very scarce.

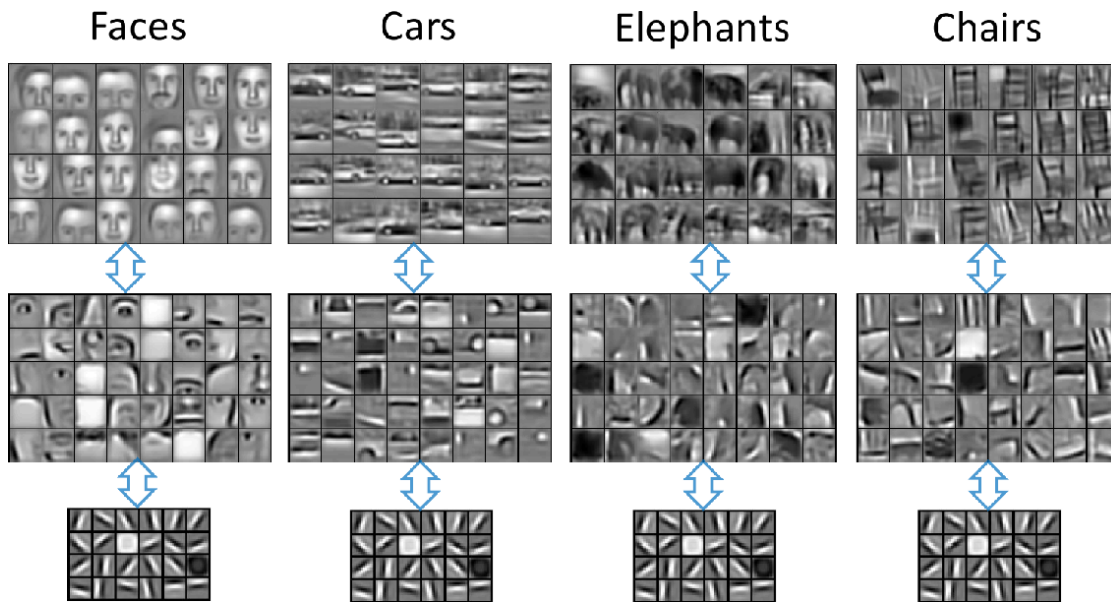


Figure 2.11: Visualization of Convolutional Kernels at different levels in Convolutional Neural Networks for different classes of images

2. **Fine Tuning:** transfer the layers from a pre-trained network and train the entire network on the available labelled data. This set-up needs a little more labelled data because the entire network is trained and hence there is a large number of parameters. This set-up is more prone to overfitting when there is a scarcity of data.

3. **Two-Stage Transfer Learning:** train the newly attached layers while freezing the transferred layers for a few epochs before fine-tuning the entire network. Fine-tuning the entire network without giving a few epochs to the final layers can result in the propagation of harmful gradients from randomly initialized layers to the base network. Furthermore, fine-tuning requires a comparatively smaller learning rate, and a two-stage approach is a convenient solution to it.

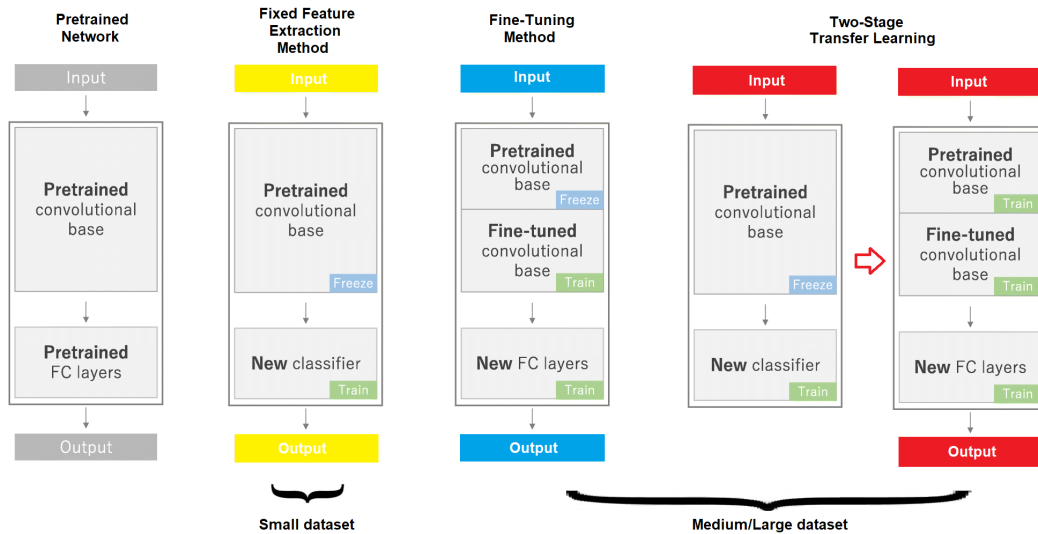


Figure 2.12: Three methods to use Transfer Learning in CNNs

2.5 Multi-Task Learning

Multi-Task Learning (MTL) is a learning paradigm in machine learning and its aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks.

At its early stage, an important motivation of MTL is to alleviate the data sparsity problem where each task has a limited number of labelled data. In the data sparsity problem, the number of labelled data in each task is insufficient to train an accurate learner and instead MTL aggregates the labelled data in all the tasks, in the spirit of data augmentation, to obtain a more accurate learner for each task. From this perspective, MTL can help reuse existing knowledge and reduce the cost of manual labelling for learning tasks.

One reason that MTL is effective is that it utilizes more data from different learning tasks when compared with single-task learning. With more data, MTL can learn more robust and universal representations, obtaining more powerful models, with better knowledge sharing among tasks, better performance on each task, and a low risk of overfitting in each task.

MTL is related to other learning paradigms in machine learning, including transfer learning and multi-label learning.

The setting of MTL is similar to that of transfer learning, but they have a significant difference. In MTL, there is no distinction among different tasks and the objective is to improve the performance of all the tasks. However, transfer learning is to improve the performance of a target task with the help of source tasks, hence the target task plays a more important role than source tasks. In a word, MTL treats all the tasks equally, instead in transfer learning the target task attracts most attentions.

From the knowledge flow perspective, flows of knowledge transfer in transfer learning are from source task(s) to the target task, instead in multi-task learning, there are flows of knowledge sharing between any pair of tasks (as shown in figure 2.13)

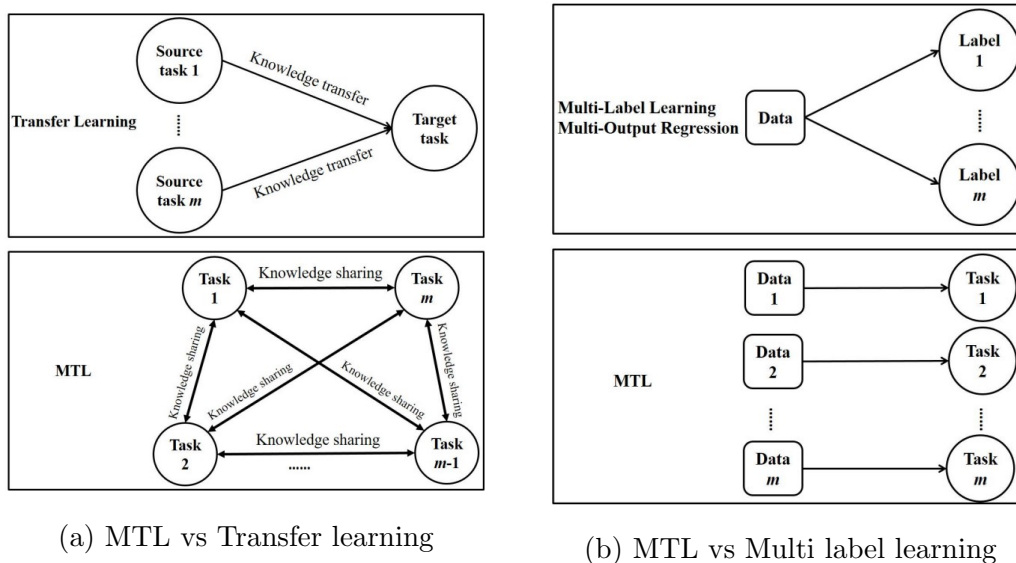


Figure 2.13: Illustrations for differences between MTL and other learning paradigms

In multi-label learning, each data point is associated with multiple labels which can be categorical or numeric. If each of all the possible labels is treated as a task, multi-label learning can be viewed in some sense as a special case of multi-task

learning where different tasks always share the same data during both the training and testing phases.

MTL is also different from continual learning in which tasks come sequentially, tasks are learned one by one, while MTL is to learn multiple tasks together, typically by learning a joint representation of the data.

Chapter 3

Background on Face Vision

This chapter presents some concepts related to the field of computer vision when dealing with face images or video. These concepts are not described in depth, but can be useful to have an overview of the field and the terminology used. This can also be helpful to understand the more complex and advanced state-of-the-art methodologies of the head pose estimation sub-field presented in the following chapters.

3.1 Face Related Computer Vision Tasks

In the last decades, a number of popular research subjects related to face have grown up in the community of computer vision, however, these tasks embrace wide and different concepts. In our industry, terms such as face detection and face recognition are sometimes used interchangeably, but there are actually some key differences.

- **Face Detection:** is a computing technology capable of locating the presence of human faces within digital images and video. It was introduced in 2001 and can be considered a subcategory of object detection technology.

Face detection does not identify people or give names to faces. The technology simply checks to see whether there is, in fact, a person in a certain photograph or video. It uses machine learning algorithms to scan digital

images for human faces, typically by looking for the eyes first and then calculating the edges of each human face. This is how the system pinpoints exactly where human faces are and counts how many people are present in photos or videos.

- **Face Recognition:** is the task of making a positive identification of a face in a photo or video image against a pre-existing database of faces. It begins with detection - distinguishing human faces from other objects in the image - and then works on the identification of those detected faces. Where face detection simply identifies the presence of a face in an image, face recognition either verifies or identifies an actual person.

Once a system establishes that there is, in fact, a face present, it uses a series of algorithms to examine that face and get details about it. These details are known as “facial landmarks” and the more landmarks a face recognition system can read, the more accurate the system will be, this specific activity is called **Face Landmark Detection**. These landmarks are used to create a precise geometrical/mathematical representation of the face and find a match with a specific person.

Typically, to obtain better performance in the recognition task once the geometric structure of a face is computed, translation, rotation and scale transformations are applied to obtain a canonical alignment of the face, for this reason, this task is also known as **Face Alignment**.

Sub-tasks of face recognition are **Face Verification** and **Face Identification**. Face verification is the task of comparing a candidate’s face to another and verifying whether there is a match. It’s a one-to-one mapping: the system checks if this person is the correct one. This can be done Frontal-Frontal, so comparing two frontal pose images, or Frontal-Profile [51].

Face identification is distinct from face verification, indeed the latter is carried out with the individual’s consent. Instead, face identification, on the other hand, scans faces and then runs them against a database to identify the people, usually on behalf of law enforcement. Individuals cannot opt out

of the process.

- **Face Analysis:** unlike face recognition, face analysis doesn't pursue identification or verification. Instead, it focuses on gathering actionable insights from facial expressions and positioning, without putting a name to the face. Face analysis can figure out where human eyes are focused and can even "read" the emotions on human faces.

A typical task is **Face Attribute Estimation**, also called Face Attribute Classification or Face Attribute Prediction, various attributes of a facial image, such that soft-biometrics traits [59] (e.g. whether someone has a beard) or whether the person is wearing a hat, and so on are detected.

Facial attribute analysis aims to build a bridge between human-understandable visual descriptions and abstract feature representations required by machine learning models. This task is strictly connected with the task of **face image synthesis** (also called face generation) which is the task of generating (or interpolating) new faces from an existing dataset.

The synthesis of new images is very important because it can be used as a data augmentation technique (described in detail in section 3.1.2) employed to cover more real-world scenarios and wide-range of attribute types, or also to solve the problem of imbalanced data distribution of facial attribute images [50].

On the other hand, face analysis can be a significant quantitative performance evaluation criterion for face synthesis models, where the accuracy gap between real images and generated images can reflect the performance of deep generative algorithms.

Finally, many face analysis tasks and face synthesis methods benefit from **Face Segmentation**, this segments the face images according to the different regions in the face (eyes, nose, ears, etc.) at a pixel-level granularity, dividing the image into different parts according to visual understanding, giving additional information that can improve the performance of many algorithms.



Figure 3.1: Example of different Computer Vision face related tasks

3.1.1 Facial Analysis

Face Analysis detects faces in images or video and then uses face tracking and action units to accurately provide information for the faces detected. It is very useful to track and respond to human behaviour in real-time, or to build engaging customer experiences and maximize their satisfaction or moreover to get insights into the effect of various stimuli and emotions.

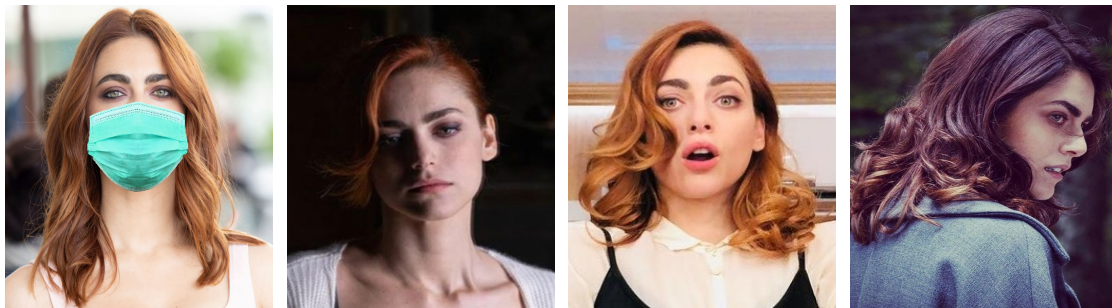
For these reasons, there are many sub-tasks that interest facial analysis and that can be solved with a multi-task model or from different algorithms separately. The most common are:

- **Facial attribute prediction** is a Computer Vision (CV) task about deducing the set of attributes belonging to a face. Examples of attributes are: *colour of hair, hairstyle, age, gender, etc.*
- **Face emotion recognition:** is the task of analysing facial expressions to reveal information on one's emotional state. The emotions are typically classified in some standard classes, such as *happy, angry, sad, neutral, surprise, disgust* or *fear*.
- **Head pose estimation:** is the task of finding the relative orientation (and position) of the human's head with respect to the camera. In particular, in the head pose estimation task, it is common to predict relative orientation with Euler angles - *pitch, yaw* and *roll*.
- **Gaze estimation:** is a task to predict where a person is looking at given the person's full face. The task contains two directions: 3-D gaze vector and 2-D gaze position estimation. *3-D gaze vector estimation* is to predict the gaze vector, which is usually used in automotive safety. *2-D gaze position estimation* is to predict the horizontal and vertical coordinates on a 2-D screen, which allows utilizing gaze point to control a cursor for human-machine interaction.

Facial analysis is a challenging task: it can involve *face localization* first and then for example *face alignment* or *face segmentation* before *attribute prediction*.

Moreover, faces are inherently difficult to analyse due to their complex appearance [49]. Indeed, the appearance can be altered, being even more complex, by **face variations**. The most common forms of face variations are the following:

- *Occlusions*: hairstyle, make-up, glasses (especially sunglasses), hats and other kinds of objects can hide meaningful pixels needed by the models to detect a face and its attributes. In case of extreme occlusion, a model may not be able to localize a face at all.
- *Illumination*: extreme lightning or extreme shadowing can make the work of a detection/analysis algorithm much harder (if not impossible) as occlusion do.
- *Expression*: emotions can alter the way a face normally appear. If a face detection, or face analysis, system has never seen faces subject to emotions during training, it can fail to detect them correctly.
- *Pose*: high-degree rotations in terms of *pitch* (x-axis), *yaw* (y-axis) and *roll* (z-axis) can eventually alter both the appearance of a face and hide its facial features.



(a) Occlusion

(b) Illumination

(c) Expression

(d) Pose

Figure 3.2: Example of face variations

Thus, it's very important that a machine learning model is trained on difficult exemplars of faces in order to generalize well to *faces captured in-the-wild* (faces captured under any kind of conditions).

Facial analysis is not only an academic challenge, but also a way to improve existing applications. For example, a photo app can detect the “smiling” attribute in order to decide which is the best photo among a given sequence; biometric capabilities can be used to unlock phones, make secure digital payments or also determine whether drivers are focused on the road.

Soft-biometric characteristics play a major role in facial research and applications [59]. Recently, there is a high interest in studying these attributes and mitigating their effects on recognition performances for fair face recognition systems, but also are important for access control, human-computer interaction, and law enforcement.

3.1.2 Data Augmentation

Deep learning strongly relies on large and complex training sets to generalize well in unconstrained settings. However, collecting and labelling a large quantity of real samples is widely recognized as a laborious, expensive and error-prone activity, and existing datasets are still lack of variations compared to the samples in the real world. Data augmentation is a valid alternative to compensate for the insufficient facial training data. Typically referred to as *face data augmentation*, this technique is used to enlarge the training or testing data size by transforming collected real face samples or simulated virtual face samples [52].

The two main advantages of face data augmentation are:

1. If controllable generation method is adopted, faces with specific features and attributes can be obtained.
2. It can generate faces without self-occlusion and balanced datasets with more intra-class variations.

At the same time, face data augmentation has some limitations:

1. The generated data can lack realistic variations in appearance, such as variations in lighting, make-up, skin colour, occlusion and sophisticated background, which means the synthetic data domain has different distribution

with respect to real data domain. That is why some researchers use *domain adaptation* and *transfer learning* techniques to improve the utility of synthetic data.

2. The creation of high-quality synthetic data is challenging. Most generated face images lack facial details, and, usually, the resolution is not high. Furthermore, some other problems are still under study, such as identity preserving and large-pose variation [52].

There are different transformation types that can be applied in face data augmentation (as shown in figure 3.3):

- **Generic transformations:** the generic data augmentation techniques can be divided into two categories: geometric transformation and photometric transformation. These methods have been adapted to various learning-based computer vision tasks. Geometric transformation alters the geometry of an image by transferring image pixel values to new positions. This kind of transformation includes translation, rotation, reflection, flipping, zooming, scaling, cropping, padding, perspective transformation, elastic distortion, lens distortion, mirroring, etc.

Photometric transformation alters the RGB channels by shifting pixel colours to new values, and the main approaches include colour jittering, grayscaling, filtering, lighting perturbation, noise adding, etc.

- **Component transformations:** the component data augmentation technique focuses on the semantic content of the image, some elements such as hairstyle, makeup and worn accessories (eyeglasses, hat, etc.) can affect face detection and recognition due to the occlusion and appearance variation of face it caused. Altering the original image by generating numerous samples with different component characteristics in the training data makes the algorithm more robust.
- **Attribute transformations:** the attribute data augmentation, similarly to the component technique, focuses on the semantic content of the input.

In this case, new images are generated by changing the soft-biometric characteristics of the face, such as age, skin colour, gender, etc.

These last two types of data augmentation are face synthesis techniques, therefore are typically based on generative models.

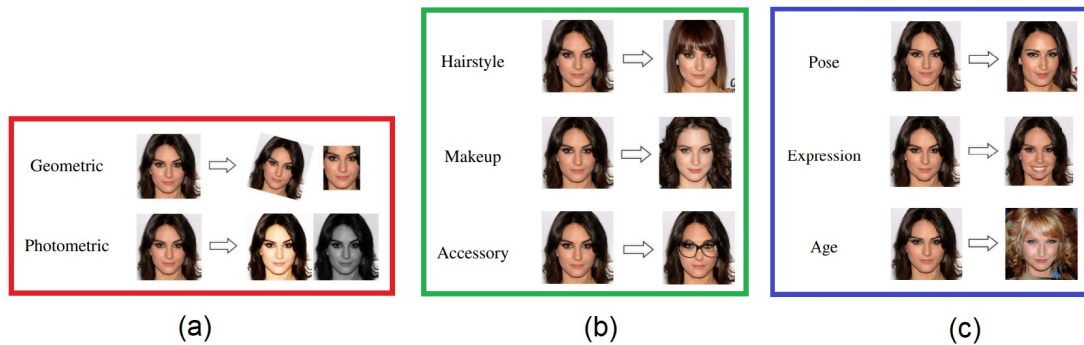


Figure 3.3: An overview of transformation types. (a) Generic transformations, (b) Component transformations, (c) Attribute transformations (image from [52])

3.2 3D Morphable Face Models

More than two decades ago, 3D Morphable Face Models (3DMM) were proposed for general face representation as well as image analysis. Today they have continued to attract considerable interest because of their ability to model intrinsic properties of 3D faces, such as shape and skin texture, rather than their appearance, with many uses in face recognition, entertainment, healthcare, forensics, computer graphics, animations and more.

The last few years seem to have re-discovered Morphable Face Models especially due to advances in deep learning and their application in state-of-the-art face analysis.

The basic 3D model consists of two components, vertexes (points in the space) and faces (triangles formed by vertexes). Vertexes define the shape of the model's faces and make the model connected. Vertexes and faces together are called *mesh*. The more vertexes in the mesh are, the more detailed the 3D object results.

Therefore a face 3DMM serves to estimate the 3D shape of a head from an image, so as to create a 3D representation of that person.

Initially, this idea of morphing started out from the core concept of isolating the core parts of a face (features) that could change from one face to another one. Finding these features was accomplished up to 2015 by using PCA (principal component analysis) on some 3D datasets of heads. By doing this a 1D vector representation of each face is created, where the value of a single feature told how far away (deformed) from the average face mesh that specific feature is.

To find the 1D vector that corresponds to a specific face, first a 3D face template is (manually) aligned over the target image (to make sure to have the correct rotation, orientation, azimuth, etc..) and then an optimisation process is executed to try to tune the values of the initial 1D representation of the template 3D face mesh, so as to minimize the RMSE between the target image and the 2D projection of the face mesh. Actually, a face can be represented by two 1D vectors S and T that represent shape and texture (face colour) respectively.

With this representation, that describes a face by a 1D vectors of values, the objective is to find the 1D vector that morphed the model face mesh into a 3D face that, when projected on a 2D surface (like when a picture was taken), would be close to the original image.

The position of the face must be known in order to make the correct 2D projection work, therefore this process is not end-to-end.

To overcome the limitations of PCA new modern approaches uses Deep Convolutional Neural Networks to learn 3DMM parameters independently and largely replaced traditional optimization based methods with more accurate results and shorter running time [164].

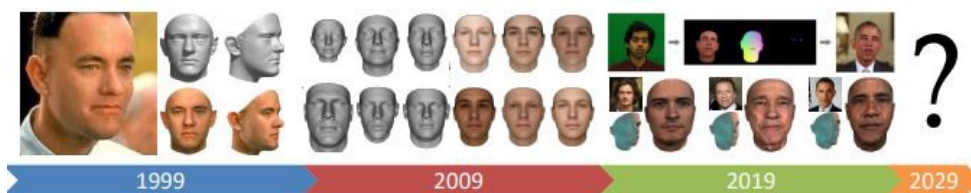


Figure 3.4: 20 Years of 3D Morphable Models (image from [53])

3.3 3D Dense Face Alignment

Face alignment in 2D aims at locating a sparse set of fiducial facial landmarks from a face image. It is called sparse because it uses a limited number of facial keypoints (68 is a common number). Instead, **3D face alignment** aims to fit a 3D morphable model (3DMM) from a 2D image, obtaining a representation with a much higher number of points (tens of thousands of points) and having the potential to deal with larger poses and occlusions [4]. 3DMM is more flexible compared to 2D models, for this reason, the 3D solution has advantages in the alignment of faces in large poses. However, the 3D solution is more complicated in modelling, fitting and data labelling [77].

A relevant, but different, problem is the **3D face reconstruction**, which recovers a 3D face from given 2D landmarks, capturing fine facial details.

In recent years, 3D face reconstruction and face alignment tasks have gradually been combined into one task: **3D dense face alignment**, which is the reconstruction of a face’s 3D geometric structure with pose information. It requires methods to offer pixel-wise facial region correspondence between two face images. 3D dense face alignment can power face-related tasks such as facial recognition, animation, facial tracking, attribute classification and image restoration [164].

Recent studies on 3DDFA are mainly divided into two categories: 3D Morphable Model (3DMM) parameters regression and dense vertices regression. *Dense vertices regression* methods directly regress the coordinates of all the 3D points (usually more than 20,000) through a fully convolutional network. The resolution of reconstructed faces however relies on the size of the feature map which in turn is based on heavy networks which are slow and memory-consuming. Compared with dense vertices, *3DMM parameters regressors* have low dimensionality and low redundancy, which the researchers regard as more appropriate to regress using a lightweight network. The regression however becomes challenging, as different 3DMM parameters influence the reconstructed 3D face differently, and parameters must be re-weighted according to their importance during training.



Figure 3.5: Example of 3D dense face alignment

Chapter 4

Head Pose Estimation

Recently, head pose estimation has become a popular area of research. Applications of HPE are wide-ranging and include (but are not limited to) virtual & augmented-reality, driver assistance, markerless motion capture or as an integral component of gaze-estimation, since gaze and head pose are tightly linked (as shown in section 1). It is also important in providing visual cues for the targets of conversation, to indicate appropriate times for speaker/listener role switches as well as to indicate agreement [61] [62].

In the context of computer vision, head pose estimation is most commonly interpreted as the ability to infer the orientation of a person's head relative to the view of a camera. More rigorously, head pose estimation is the ability to infer the orientation of a head relative to a global coordinate system, but this subtle difference requires knowledge of the intrinsic camera parameters to undo the perceptual bias from perspective distortion [55].

At the coarsest level, head pose estimation applies to algorithms that identify a head in one of a *few discrete orientations*, e.g., a frontal versus left/right profile view. At the fine (i.e., granular) level, a head pose estimate might be a *continuous angular measurement* across multiple Degrees of Freedom (DoF). A system that estimates only a single DoF, perhaps the left to right movement, is still a head pose estimator, as is the more complex approach that estimates a full 3D orientation and position of a head.



(a) Coarse pose estimation

(b) Fine pose estimation

Figure 4.1: Example of coarse and fine head pose estimation

In particular, in the head pose estimation task, it is common to predict relative orientation with Euler angles - *pitch*, *yaw* and *roll*. They define the object's rotation in a 3D environment, if the right prediction about these three angles can be made, it can be found in which direction the human head will be facing.

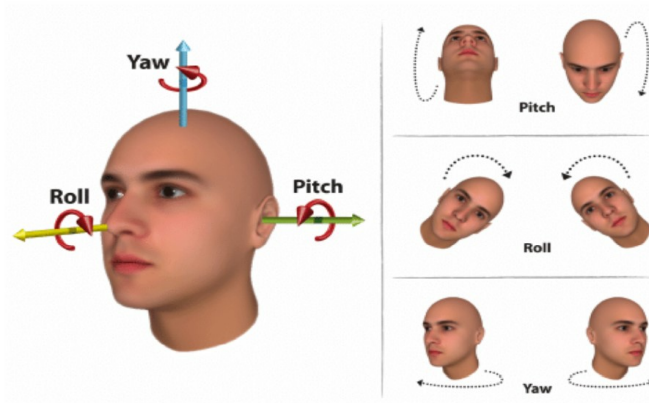


Figure 4.2: Euler angles in Head Pose Estimation. The rotation occurring around the axis passing from the head through the neck is called *yaw*. The rotation occurring around the axis passing through the ears is named *pitch*. The rotation occurring around the axis connecting the nose with the occipital bone is known as *roll* (image source [54]).

Despite the head pose estimation task may seem to be easily solved, achieving acceptable quality on it has become possible only with recent advances in Deep Learning.

Challenging conditions like extreme pose, bad lighting, occlusions and other faces in the frame make it difficult for data scientists to detect and estimate head poses (examples in figure 3.2).

Despite this SOTA methods for head pose estimation satisfy all the following criteria, firstly proposed by Erik Murphy-Chutorian in [55], on standard datasets:

- **Accurate:** the system should provide a reasonable estimate of the pose with a mean absolute error of 5° or less.
- **Monocular:** the system should be able to estimate head pose from a single camera. Although accuracy might be improved by stereo or multi-view imagery, this should not be a requirement for the system to operate.
- **Autonomous:** there should be no expectation of manual initialization, detection, or localization, precluding the use of pure-tracking approaches that measure the relative head pose w.r.t. some initial configuration and shape/-geometric approaches that assume facial feature locations are already known.
- **Multi-Person:** the system should be able to estimate the pose of multiple people in one image.
- **Identity & Lighting Invariant:** the system must work across all identities with the dynamic lighting found in many environments.
- **Resolution Independent:** the system should apply to near-field and far-field images with both high and low resolution.
- **Full Range of Head Motion:** the methods should be able to provide a smooth, continuous estimate of *pitch*, *yaw* and *roll*, even when the face is pointed away from the camera.
- **Real-Time:** the system should be able to estimate a continuous range of head orientation with fast (30fps or faster) operation.

4.1 Datasets

In order to truly make progress in the problem of predicting pose from image intensities, real datasets which contain precise pose annotations, numerous identities, different lighting conditions, all of this across large poses occur.

Most of the HPE models are evaluated using publicly available datasets. These datasets significantly evolved during the last years, especially in terms of complexity of environmental conditions.

Most datasets provide rotation information by means of Euler angles, which define the orientation of a rigid body with respect to a fixed coordinate system. Accordingly, three rotations are always sufficient to reach any target position. These rotation angles can be extrinsic or intrinsic, the former expresses the rotations with respect to the xyz axes of an original motionless coordinate system, the latter expresses rotations with respect to axes of a rotating XYZ coordinate system, rigidly attached to the moving body.

Since various formalisms exist to express a rotation in three dimensions beyond Euler angles e.g., rotation matrices, unit quaternions, Rodrigues' formula, among others, the datasets contain different forms of representation (many of these formalisms use more than the minimum number of three parameters). More details about some of the representations exploited by the models to solve the HPE task can be found in section 4.2.

Head pose datasets can be categorized by different aspects, such as imaging characteristics, data diversity, acquisition scenario, annotation type, and annotation technique [32]. These aspects play an important role on whether and how the dataset identifies the challenges of the head pose estimation task.

- Imaging characteristics: relate to the image resolution, number of cameras, bit depth, frame rate, modality (RGB, grayscale, depth, infrared), geometric setup and field of view.
- Data diversity: incorporates aspects such as the number of subjects, the distribution of age, gender, ethnicity, facial expressions, occlusions (e.g. glasses,

hands, facial hair) and head pose angles. Data diversity is essential to training and evaluating robust estimation models.

- **Acquisition scenario:** covers the circumstances under which the acquisition of the head pose takes place. The most important distinction is between *in-laboratory* vs. *in-the-wild* acquisition. While the former restricts the data by defining a rather well-defined, static environment, the latter offers more variety through being acquired in unconstrained environments, such as outside, thus covering many challenging conditions like differing illumination and variable background. Head movement can be staged by following a predefined trajectory or can be naturalistic by capturing head movement while the subject performs a different task, such as driving a car.
- **Annotation type:** describes what meta-information, such as head pose, comes alongside the image data and how it is represented. For example, the head pose can be defined by a full 6 degrees of freedom (DoF) transformation from the camera coordinate system to the head coordinate system (covering 3 DoF for translation and 3 DoF in rotation) or only a subset of them can be provided. Annotation types can differ also in their granularity of sampling the DoF space: there are discrete annotation types that classify a finite set of head poses, and there are continuous annotation types that offer head pose annotations on a continuous scale for all the DoFs.
- **Annotation technique:** there are different methods for obtaining the head pose annotation (label) accompanying each image. The annotation technique has a large impact on data quality. These are described in more detail in section 4.3.

Database	Year	# sub- jects	# images	Yaw	Pitch	Roll	DB type	GT method	Pose type
BU [30]	2000	5	200	✓	✓	✓	C	MS	C
PIE [39]	2000	68	40.000	✓			C	CA	D
IDIAP-HP [31]	2003	16	66.295	✓	✓	✓	C	MS	C
CAS-PEAL [19]	2004	1.040	99.594	✓	✓		C	CA	D
Pointing'04 [29]	2004	15	2.790	✓	✓		C	DS	D
FacePix [28]	2005	30	5.430	✓			C	CR	D

Bosphorus [25]	2008	105	4.652	✓	✓		C	DS	D
ETH [27]	2008	26	10.000	✓	✓		C	ICP	C
BJUT-3D [26]	2009	500	46.500	✓	✓		C		
Taiwan RoboticsLab [40]	2009	90	6.660	✓			C	CA	D
Multi-Pie [24]	2010	337	75.000	✓			C	CA	D
AFLW [23]	2011		25.993	✓	✓	✓	W	E	C
BIWI Kinect [5]	2011	20	15.000	✓	✓	✓	C	ICP	C
AFW [22]	2012	205	468	✓	✓	✓	W	M	D
ICT-3DHP [21]	2012	10	1.400	✓	✓	✓	C	IS	C
BioVid Heat Pain [36]	2013	90	9.000	✓	✓	✓	C	ICP	C
CAVE [38]	2013	56	5.880	✓			C	CA	D
McGill [20]	2013	60	18.000	✓			W	M	D
Dali3DHP [17]	2014	33	60.000	✓	✓	✓	C	IS	C
MTFL [107]	2014		12.995	✓			W	M	D
300W-LP [4]	2015		122.450	✓	✓	✓	H (W+S)	S	C
AFLW2000-3D [4]	2015		2.000	✓	✓	✓	W	E	C
AISL [18]	2015	20	6.480	✓	✓		C	CR [†]	D
CMU Panoptic [◊] [8]	2015		1.342.018	✓	✓	✓	C	P	C
CCNU [15]	2016	58	4.350	✓	✓		C	IS	C
GI4E-HP [14]	2016	10	36.000	✓	✓	✓	C	MS	C
Synthetic [16]	2016	37	74.000	✓	✓	✓	S	S	C
UMDFace [6]	2016	8.277	367.888	✓	✓	✓	W	E	C
DriveAHead [35]	2017	20	~ 1 M	✓	✓	✓	W*	O	C
Pandora [7]	2017	22	250.000	✓	✓	✓	C*	IS	C
SASE [12]	2017	50	30.000	✓	✓	✓	C	ICP	C
SyLaHP [41]	2017	30	~ 101 K	✓	✓	✓	S	S	C
SynHead [11]	2017	10	510.960	✓	✓	✓	S	S	C
UbiPose [10]	2018	22	10.400	✓	✓	✓	C	ICP	C
VGGFace2 [3]	2018	9.131	~ 3,31 M	✓	✓	✓	W	E	C
DD-Pose [32]	2019	27	~ 330 K	✓	✓	✓	W*	O	C
GOTCHA-I [42]	2019	62	137.826	✓	✓	✓	W	E	D
M2FPA [9]	2019	229	397.544	✓	✓		C	CA	D
AutoPOSE [33]	2020	20	1.018.885	✓	✓	✓	C*	O	C
MDM corpus [34]	2021	59	~ 10,5 M	✓	✓	✓	W*	ICP	C
UET-Headpose [2]	2021	9	12.848	✓	✓	✓	C	IS	C

Database:

- \diamond = Processing operations needed to extract head pose information from original data [61]

DB Type:

- C = Constraint, faces of real people taken in a constraint environment (a lab, an office, etc.)
- W = In-the-Wild, images of real people captured under any kind of conditions
- S = Synthetic, synthetic generated images
- H = Hybrid, a mixture of previous types
- * = Dataset build for the driving context

Pose Type:

- C = Continuous, pose estimate in continuous range
- D = Discrete, few discrete orientations are acquired

GT Type:

-
- CA = Camera array
 - CR = Camera ring
 - CR[†] = It's not the camera that rotates around the person, but the seat that rotates on itself
 - DS = Directional suggestion
 - E = Estimation with neural networks or other algorithms
 - ICP = ICP algorithm
 - IS = Inertial sensor
 - L = Laser pointer directional suggestion
 - M = Manual annotation
 - MS = Magnetic sensor
 - O = Optical motion capture system
 - P = Panoptic studio
 - S = Synthetic images generation
-

Table 4.1: Available datasets for Head Pose Estimation. The most used in the literature are in bold.

There are many available datasets in the literature:

- **300W-LP** [4]: The 300W-LP (Large Pose) is a synthetic extension of the 300W database [37], generated to augment the number of challenging samples with extreme poses. It includes 122.450 images with yaw angle in range $\pm 89^\circ$.
- **AFLW** [23]: Annotated Facial Landmark in the Wild is a challenging dataset which was collected from the internet, in totally unconstrained conditions. It contains a collection of 25.993 faces with head poses ranging between $\pm 120^\circ$ for yaw and $\pm 90^\circ$ for pitch and roll. The pitch, yaw and roll angles were obtained automatically from the labelled landmarks using the POSIT algorithm [75], assuming the structure of a mean 3D face, for this reason, several annotations errors were found [172].
- **AFLW2000-3D** [4]: This dataset contains the first 2.000 identities of the in-the-wild AFLW [23] dataset which have been re-annotated with 68 3D landmarks using a 3D model which is fit to each face. Consequently, this dataset contains accurate fine-grained pose annotations and is a prime candidate to be used as a test in head pose estimation task. Yaw varies $\pm 120^\circ$, while roll and pitch $\pm 90^\circ$.

- **AFW** [22]: Annotated Faces in the Wild represents a small database (it's a subset of AFWL [23]), which is normally used for testing purposes only. AFW has 250 images and inside these images 468 faces in a very challenging environment are included. The yaw angles vary between $\pm 90^\circ$ with a step size of 15° . The ground-truth is manually annotated, so it may contain errors.
- **AISL** [18]: The Aisl head orientation database is a collection of small scale head images with various backgrounds of an indoor scene. This dataset contains 6.480 images of 20 subjects under 36 yaw angles, 3 pitch angles and 3 different backgrounds. The orientation is determined by two categories: yaw angle in 360° with an interval of 10° , and pitch angle in the range $\pm 45^\circ$ with an interval of 45° .
- **AutoPOSE** [33]: It's a large-scale dataset that provides 1.1 million images taken from a car's dashboard view. AutoPOSE's ground-truth head orientation was acquired with a sub-millimetre accurate motion capturing system placed in a car simulator. The rotations are limited to the range $[-90^\circ, +90^\circ]$, the average pitch angle is shifted in the negative values of the rotation angles, this is due to the placement of the camera in the dashboard.
- **BioVid Heat Pain** [36]: It contains videos and physiological data of 90 persons subjected to well-defined pain stimuli of 4 intensities, built for the development of automatic pain monitoring systems. It includes information about head pose of the recorded subjects for all 3 angles pitch, yaw, roll, all in the range $\pm 50^\circ$.
- **BIWI Kinect** [5]: It's gathered in a laboratory setting by recording RGB-D video of different subjects across different head poses, using a Kinect v2 device. It contains roughly 15.000 frames and the rotations are $\pm 75^\circ$ for yaw, $\pm 60^\circ$ for pitch and $\pm 50^\circ$ for roll. A 3D model was fit to each individual's point cloud and the head rotations were tracked to produce the pose annotations. This dataset is commonly used as a benchmark for pose estimation using depth methods that attest to the precision of its labels.

- **BJUT-3D** [26]: The database consists of 46,500 images collected from the 3D faces of 250 male and 250 female participants. The total number of poses in the database is 93. The pitch rotation is quantized into 9 angles $[-40^\circ, +40^\circ]$, where the difference between two consecutive poses is 10° . Similarly, the yaw rotation is divided into 13 angles $[-60^\circ, +60^\circ]$, with the same angular step size as for the pitch.
- **Bosphorus** [25]: It contains 5 thousand high resolution face scans from 105 different subjects. The 3D scans are obtained by a commercial structured-light based 3D digitizer. It offers 13 discrete head pose annotations (seven yaw angles, four pitch angles, and two roll angles), with different facial expressions and occlusions.
- **BU** [30]: The Boston University Head Tracking dataset includes only 200 images and 5 subjects, which is the main drawback of this database. The acquisition process is repeated in two sessions: initially illumination conditions are uniform; then subject faces are exposed to rather complex scenarios with changing illumination. All three rotation angles were recorded thanks to a magnetic tracker attached to each participant's head. Pose variation is mainly less than 30° . Since the presence of facial occlusions (e.g., eyeglasses, facial hair, etc.) is very limited, most methods perform very well.
- **CAS-PEAL** [19]: The CAS-PEAL is a large dataset having 99,594 images, with a total number of 1,040 participants, with 595 males and 445 female subjects. The CAS-PEAL dataset contains a total of 21 poses combining different yaw and pitch angles: the yaw orientation varies between -45° and $+45^\circ$ with an interval of 15° between two consecutive poses; the pitch orientation has only three poses -30° , 0° and $+30^\circ$. Although the dataset has sufficient data for evaluation and training, its complexity is low, as the number of poses is quite limited.
- **CAVE** [38]: The Columbia Gaze dataset contains a total of 5,880 images of 56 different subjects (32 male, 24 female) of different ethnic groups and

ages. The dataset is mainly created to solve the gaze estimation task, but contains also information about head pose of the participants, therefore it can be used to solve the discrete head pose estimation task. For each subject a combination of five horizontal head poses (0° , $\pm 15^\circ$, $\pm 30^\circ$), seven horizontal gaze directions (0° , $\pm 5^\circ$, $\pm 10^\circ$, $\pm 15^\circ$), and three vertical gaze directions (0° , $\pm 10^\circ$) are available.

- **CCNU** [15]: All images in CCNU are low-resolution images collected in a classroom. The database consists of 58 participants, captured in 75 different poses, for a total number of 4.350 images. The face images are collected so that illumination conditions and facial expressions are changing, thus adding more complexity to the images. For obtaining the ground-truth data, SensoMotoric Instruments (SMI) eye tracking glasses are used. The head orientation changes from -90° to $+90^\circ$ in the horizontal direction, while the vertical direction spans in the range -45° to $+90^\circ$.
- **CMU Multi-Pie** [24]: This is a database collected from subjects exhibiting multiple expressions under different illumination conditions in a constraint environment. All high-resolution images are captured using a system of 15 cameras for a total of 75 thousand images. The only angle of rotation available is the yaw with an incrementation step of 15° .
- **CMU Panoptic Dataset** [8]: It's a large scale dataset providing 3D pose annotations for multiple people engaging social activities. It contains 65 videos with multi-view annotations captured inside a dome from approximately 30 HD cameras. The panoptic dataset includes 3D facial landmarks and calibrated camera extrinsics and intrinsics, but does not include head pose information. By using landmarks and camera calibrations it is possible to locate and crop images of subjects' heads and compute the corresponding camera-relative Euler angles.

After processing the dataset to address the head pose problem [61], it contains 1.342.018 images. The yaw angle distribution is almost uniform and ranges in $\pm 179^\circ$, but at angles near 90° and -90° there are fewer images due

to the effect of Gimbal lock. For the two angles pitch and roll the magnitudes are in the range $\pm 89^\circ$.

- **CMU-PIE** [39]: The CMU Pose, Illumination, and Expression (PIE) dataset contains over 40.000 facial images of 68 people. Using the CMU 3D Room each person is imaged across 13 different poses, under 43 different illumination conditions and with 4 different expressions. The pose ground-truth was obtained with a 13 cameras array, each positioned to provide a specific relative pose angle. This consisted of 9 cameras at approximately 22.5° intervals across yaw, one camera above the centre, one camera below the centre, and one in each corner of the room.
- **Dali3DHP** [17]: This database is an extreme head pose database collected from a camera mounted on a treadmill. The dataset was collected in two different sessions from 33 individuals. Ground-truth data is collected using Shimmer sensor 2 which was attached to each person’s head. The database is large since it contains more than 60.000 depth and colour images. All the three rotation angles pitch, yaw and roll were defined at the time the acquisition took place, covering the following head angles: pitch $[-65.76^\circ, +52.60^\circ]$, roll $[-29.85^\circ, +27.09^\circ]$, and yaw $[-89.29^\circ, +75.57^\circ]$.
- **DD-Pose** [32]: It contains 330 thousand measurements from multiple cameras acquired by an in-car setup during naturalistic drives by 27 subjects. Large out-of-plane head rotations and occlusions are induced by complex driving scenarios, such as parking and driver-pedestrian interactions. Precise continuous 6 DoF head pose annotations are obtained by a motion capture sensor and a novel calibration device. The angles vary in the following ranges, ignoring outliers with less than 10 measurements in a 3° neighbourhood: pitch $\in [-69^\circ, +57^\circ]$, yaw $\in [-138^\circ, +126^\circ]$, roll $\in [-63^\circ, +60^\circ]$.
- **DriveAHead** [35]: It’s another driver head pose dataset, it contains frame-by-frame head pose labels obtained from a motion-capture system for 20 subjects (about 1 million of frames). It includes parking manoeuvres, driv-

ing on the highway and through a small town, different occlusions and illuminations, thus providing distributions of head orientation angles and head positions which are typical for naturalistic drives. Images were collected with a resolution of 512×424 pixels, 6 DoF, the range of angles is $[-45^\circ, +45^\circ]$ for pitch, $[-40^\circ, +40^\circ]$ for roll and mainly $[-90^\circ, +90^\circ]$ for yaw.

- **ETH** [27]: The ETH Face Pose Range Image Dataset contains more than 10 thousand images of 20 persons (3 of them being female) at a resolution of 640×480 pixels. Each person freely turned her head while the scanner captured range images at 28 fps. Yaw varies between -90° to $+90^\circ$, pitch between -45° to $+45^\circ$, whereas roll is not considered.
- **FacePix** [28]: The FacePix database is built depicting 30 individuals, for a total number of 5.430 images. It is an imbalanced dataset with 25 males and 5 females. Yaw rotation varies from -90° (extreme left profile) to $+90^\circ$ (extreme right profile), with a step size of 2° ; no other rotation angles were considered.
- **GI4E-HP** [14]: It contains 36 thousand images from 10 subjects recorded with a web-cam in an in-laboratory environment. Head pose annotations are given in 6 DoF using a magnetic reference sensor. All transformations and camera intrinsics are provided. Head pose annotations are given relative to an initial subjective frontal pose of the subject.
- **GOTCHA-I** [42]: This dataset is a collection of 682 videos of 62 subjects in 11 different indoor and outdoor environments to address both security and surveillance problems. To obtain ground-truth a 3D head model is reconstructed and elaborated using Blender software. There are 137.826 labelled frames with 2.223 head pose per subject in the range of $[-40^\circ, +40^\circ]$ in yaw, $[-30^\circ, +30^\circ]$ in pitch and $[-20^\circ, +20^\circ]$ in roll, with a step of 5° .
- **ICT-3DHP** [21]: It's a large dataset which was collected in-the-wild, i.e., captured in an unconstrained environment. All images were acquired through

the Polhemus Fastrack¹ flock of birds tracker attached to a cap the participants that contains a magnetic sensor, so that the dataset contains both RGB and depth data. The database is evaluated for all three rotation angles including pitch, yaw and roll. No accurate information about the angle ranges is provided.

- **IDIAP Head Pose** [31]: It contains 66.295 head images stemmed from a 8 video meeting recording, each approximately one minute in duration, of a few people in a meeting room. In each sequence, two subjects, which are always visible, were continuously annotated using a magnetic sensor. Therefore, each image has a complete annotation of a head pose orientation from pitch (range $[-60^\circ, +15^\circ]$), yaw (range $\pm 60^\circ$) and roll (range $\pm 30^\circ$) angles.
- **M2FPA** [9]: This dataset totally involves 397.544 images of 229 subjects with 62 poses (including 13 yaw angles, 6 pitch angles and 44 yaw-pitch angles), 4 attributes and 7 illuminations. There are 6 classes for pitch in the range of $[-30^\circ, +45^\circ]$ with a step increment of 15° and 13 measurements for yaw in the range $\pm 90^\circ$ with a step increment of 15° .
- **McGill** [20]: The database consists of 60 videos of 60 different participants, in total it contains 18.000 video frames. The videos were recorded in both indoor and outdoor environments. The participants were free to behave as they want during the video collection process, therefore arbitrary illumination conditions and background clutter are present, especially outdoor. Only yaw angles are estimated using a semi-automatic procedure, with variation in the range $[-90^\circ +90^\circ]$.
- **MDM corpus** [34]: The Multimodal Driver Monitoring database was collected with 59 subjects recorded while are driving a car and performing various tasks. To record the head pose the Fi-Cap device was used, this continuously tracks the head movement of the driver using fiducial markers, providing frame-based annotations to train head pose algorithms in naturalistic

¹<https://polhemus.com/motion-tracking/all-trackers/fastrak>

driving conditions. This set consists of 48.9 hours of recordings (10.541.166 frames), it covers a large range of head poses along all three rotation axes due to the large number of subjects included, and the variety of primary and secondary driving activities considered during the data acquisition. Yaw angles range around the origin spanning between -80° to 80° , pitch angles have an asymmetric range spanning from -50° to 100° .

- **MTFL** [107]: The Multi-Task Facial Landmark dataset contains 12.995 outdoor face images from the web. These images are from CUHK Face Alignment database and AFLW dataset. Each image is annotated with a bounding box and five facial landmarks. There are ground-truth annotations for gender, age, smiling, wearing glasses and head pose. For the latter, the images are manually categorized in 5 discrete classes: Left-profile, Left, Frontal, Right, Right-profile.
- **Pandora** [7]: It has been specifically created for head centre localization, head pose and shoulder pose estimation and is inspired by the automotive context. A frontal fixed device acquires the upper body part of the subjects, simulating the point of view of the camera placed inside the dashboard. Subjects also perform driving-like actions, such as grasping the steering wheel, looking to the rear-view or lateral mirrors, shifting gears and so on. Pandora contains more than 250 thousand full resolution RGB (1920×1080 pixels) and depth images (512×424) acquired with a Microsoft Kinect 1 device. Subjects perform wide head movements: $\pm 70^\circ$ roll, $\pm 100^\circ$ pitch and $\pm 125^\circ$ yaw. Garments as well as various objects are worn or used by the subjects to create head occlusions. The ground-truth annotations have been collected using a wearable Inertial Measurement Unit (IMU) sensor.
- **Pointing'04** [29]: It is one of the oldest databases, released in 2004, which was considered as the classical benchmark for HPE (in some studies is also called PRIMA database [74]). Despite its age, it's still used for research purposes, due to its challenging nature and a large variety in consecutive poses [96–99]. A total number of 15 participants (between 15-40 years) were

involved for image acquisitions. Some of them wear eyeglasses or show facial hairs, thus increasing the task complexity. Images were collected in an indoor lab environment, with very low illumination conditions. Each participant is asked to look at some markers on the wall, and two rotation angles (yaw and pitch) are annotated through a subsequent manual labelling process (thus introducing some errors). The head orientation varies between $\pm 90^\circ$ both in the horizontal and vertical directions, while the difference between two consecutive poses in horizontal and vertical orientation is kept at 15° and 30° , respectively.

- **SASE** [12]: This is a 3D database collected through Kinect 2 camera. It consists of both RGB and depth images of 32 male and 18 female subjects. The total number of frames is 30.000. All subjects have different ethnicity and hairstyles, with an age range of 7-35 years. All three rotation angles pitch, yaw, and roll are considered. All participants have different facial expressions during image acquisition, so that, along with head pose estimation, the database may also be used for emotion recognition. For each person a large sample of head poses are included, within the bounds of yaw from -45° to 45° , pitch -75° to 75° and roll -45° to 45° of rotation around each axis.
- **SyLaHP** [41]: The Synthetic dataset for Landmark based Head Pose estimation was proposed by Werner et al. [41] along with a benchmark protocol to learn head pose on top of any landmark detector (called HPFL). It contains about 101 thousand synthetic images from 30 subjects, with varying ethnicity, age and gender. The angles are in the ranges: $\pm 70^\circ$ for pitch, $\pm 90^\circ$ for yaw and $\pm 55^\circ$ for roll.
- **SynHead** [11]: This is a large-scale synthetic dataset for head pose estimation in videos containing 10 head models (5 female and 5 male), 70 motion tracks and 510.960 frames. Such synthetic dataset, which considers all Euler angles, generates 100% reliable ground-truth to compensate for errors existing in manually annotated datasets. The Euler angles are in the range of $[-100^\circ, +100^\circ]$.

- **Synthetic** [16]: The Synthetic image database is a large database of 74,000 high quality images taken from head models. A total of 37 sequences have been considered, where each sequence includes 2,000 frames. The head pose in face images covers $\pm 50^\circ$ of roll, $\pm 75^\circ$ for yaw, and $\pm 60^\circ$ for pitch. The database is quite challenging as different ages, races, and facial expressions are included.
- **Taiwan RoboticsLab** [40]: It contains 6,660 images of 90 subjects. Each subject has 74 images, where 37 images were taken every 5 degrees from right profile (defined as $+90^\circ$) to left profile (defined as -90°) in the yaw rotation using camera array. The remaining 37 images are generated (synthesized) by the existing 37 images using commercial image processing software in the way of flipping them horizontally.
- **UbiPose** [10]: This dataset relies on videos from the UBImpressed dataset, which has been captured to study the performance of students from the hospitality industry at their workplace. The data are recorded using a Kinect 2 sensor, however, the ground-truth head pose is indirectly inferred from facial landmarks. The validated inferred head poses are 10.4K, most frames fall within a $[20^\circ, 40^\circ]$ interval.
- **UET-Headpose** [2]: The UET-Headpose dataset was created to capture the head pose of annotated people in many conditions, it includes 12,848 images obtained from 9 people. The dataset has a uniform yaw angle distribution for all directions in the range $[-179^\circ, 179^\circ]$. The dataset is obtained by having the annotated people rotated all yaw directions when collecting the dataset. Therefore, it is possible to learn all yaw angles within a 360° range.
- **UMD Faces** [6]: This dataset has 367,888 annotated faces of 8,277 subjects. It contains information about bounding boxes (verified by humans), twenty-one keypoint locations, Euler angles and the gender of the subject. These annotations have been generated using the All-in-one CNN model [109], therefore the dataset may contain erroneous annotations, especially for the pitch, yaw and roll angles.

- **VGGFace2** [3]: This is a very large HPE database which has been released in 2018. It contains 3,31 million images. The total number of participants to create this content are 9,131, whereas the average number of images per subject is 362,6. The database is constructed with images downloaded from Google Image Search and show large variations in pose, illumination, age, profession, and ethnicity. However, pose (pitch, yaw and roll) is estimated using pre-trained pose classifiers defining 5 classes for angles in ranges $[-100^\circ, -40^\circ)$, $[-40^\circ, -10^\circ)$, $[-10^\circ, +10^\circ)$, $[+10^\circ, +40^\circ)$ and $[+40^\circ, +100^\circ)$.

4.2 Head pose rotations representations

Many possible representations can be used to express rotations of rigid bodies. The widely used in the field of head pose estimation is that based on Euler angles, but other methods are exploited in the literature due to some problems of this specific representation.

Furthermore, it has been shown that any rotation representation in 3D with less than five dimensions is discontinuous, making the learning process harder [79]. We will further briefly review different rotation parametrizations, their pros and cons to see how they might affect the regression performance.

4.2.1 Euler angles

The Euler angles were introduced by Leonhard Euler in rigid body dynamics to describe the orientation of a reference system attached to a rigid solid in motion. Three parameters are needed to describe an orientation in a 3 dimensional Euclidean Space \mathbb{R}^3 .

Thus, the Euler angles are a set of three angular coordinates which specify the orientation of a reference system with orthogonal axes, usually mobile, with respect to another reference with known orthogonal axis called standard orientation. This standard initial orientation is normally represented by a motionless (fixed) coordinate system.

Euler angles can represent any rotation by means of three successive elemental rotations around three independent axes.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These three elemental rotations around distinct axes can be composed to obtain a single rotation matrix using matrix multiplication:

$$R = R_x R_y R_z$$

Matrix multiplication is not commutative and the same thing applies to rotations, therefore the order of application of the three successive elemental rotation is important.

However, the definition of Euler angles is not unique, in the literature many different conventions are used, where varies the sequences of rotations and the axes about which the rotations are carried out.

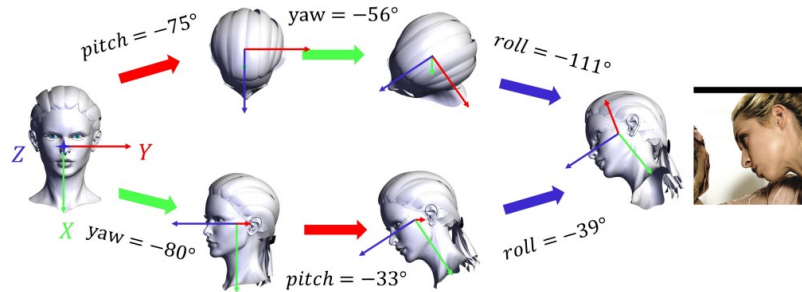


Figure 4.3: Different processes from the same initial pose to the same final pose in different rotation order (image from [208]).

Following the Trait-Bryan convention we can define as x , y and z the original axes and X , Y , and Z the axes after rotation. The line that represents the intersection between plane xy and YZ is called the line of nodes N , see figure 4.4. The Euler angles with this convention are: α the rotation angle between x and N ,

covering a range of 2π ; β the rotation angle between z and Z , covering a range of π ; γ the rotation angle between N and X , covering a range of 2π .

Many datasets have annotations of pitch, yaw and roll angles, but not all of them explicitly mention the order; the process of determining it become tedious and error-prone.

The main limitation of the Euler angles remains the **Gimbal lock**: when the second elemental rotation reaches 90 (or -90) degrees, then first and third axes become parallel (i.e. linearly dependent), which gives an infinite number of solutions for the same rotation and the other axis can not be determined. This is a great limitation when wide ranges of rotations $[-180^\circ, +180^\circ]$ are considered.

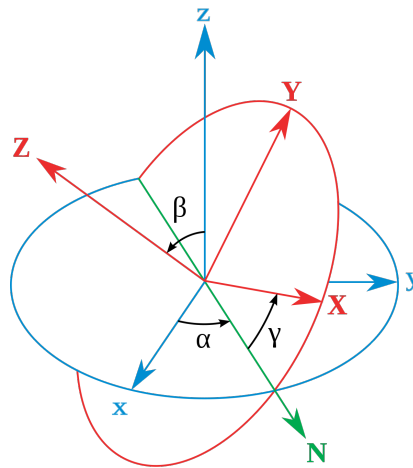


Figure 4.4: Euler angles, image from Wikipedia [66]

4.2.2 Rotation matrix

Each rotation can be uniquely described with a rotation matrix. The rotation matrix R is a special orthogonal 3×3 matrix, with a determinant equal to one, that represents a rotation in Euclidean space.

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, R^T R = R R^T = I, \det(R) = 1$$

Rotations can be composed using multiplication, and the resulting matrix will remain a rotation matrix. A rotation is represented using *nine* parameters.

To regress the parameters with back-propagation an orthogonality constraint must be enforced, otherwise something different from rotation matrix will be obtained during inference [78].

A complaint of rotation matrices is that they're less intuitive. In general, it's not easy to understand what the matrix is doing by simply looking at the matrix. This is why Euler angles sometimes are more favourable.

Let be the column vector v , the position of each point in the standard initial orientation and R the rotation matrix. Then, a rotated vector u is obtained by multiplying the rotation matrix with the vector.

$$u = R \cdot v$$

The ease by which vectors can be rotated using a rotation matrix, as well as the ease of combining successive rotations, make the rotation matrix a useful and popular way to represent rotations, even though it is less concise than other representations [58].

4.2.3 Quaternions

Quaternions are a compact way to represent rotations, they have four parameters, which can be interpreted as a scalar component plus a three-dimensional vector component:

$$q = (s_0, \vec{v}) = (s_0, v_1, v_2, v_3)$$

Quaternions are quite popular because are more compact than matrix representation and it's simple to combine two individual rotations represented as quaternions using quaternion product.

Unlike Euler angles, quaternions are free from the Gimbal lock problem, but still they have an ambiguity caused by their anti-podal symmetry: q and $-q$ correspond to the same rotation.

Furthermore, it has been recently demonstrated that for 3D rotations, all representations are discontinuous in the real Euclidean spaces of four or fewer di-

mensions and empirical results suggest that continuous representation outperform discontinuous ones [79]. This means that Euler angles and quaternions representations might not be well suited for the regression task.

4.3 Creating Ground-Truth data

The images that compose HPE datasets can be collected through different methods, in this section various methodologies for ground-truth creation are discussed.

Due to difficulties in ground-truth collection and annotation, head pose datasets can contain annotation errors and noisy data. Sources of errors may be related to inappropriate behaviour of the participants at time of acquisition (e.g. participants who do not correctly look at the suggested direction, the change in head position when different poses are acquired in subsequent moments, etc.). The acquisition sensor may also affect data quality (e.g., magnetic sensors vs. camera arrays).


In such complex acquisition scenarios, a valid alternative for training and evaluating a head pose estimation framework is constituted by synthetic datasets, where the chances of errors are comparatively less with respect to those acquired in more realistic set-ups.

Acquisition methods:

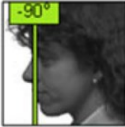
- *Manual annotation by a human*: It consists in a process by which head pose images are viewed by a human who assigns a specific label, according to his/her personal perception about the pose. This methodology has been used in the case of coarse sets of poses, typically in a single DoF, however, it is applicable only in the case of small databases and it becomes inappropriate for fine pose estimation, since the probability of human error in this case is high.
- *Generate synthetic images*: A simple way to generate training and testing data, with nearly perfect ground-truth, is to process head poses synthetically.

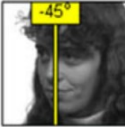
HEAD POSE (YAW ANGLE) LABELING MODULE

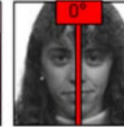
FOR THIS FACE IMAGE, PLEASE DO THE FOLLOWING TWO STEPS.
UPON FINISHING THE SECOND STEP, THIS FACE IMAGE WILL CHANGE AUTOMATICALLY.

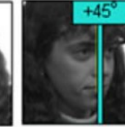



STEP 1 CLICK ON ONE OF THE IMAGES WHICH EXHIBITS THE MOST SIMILAR HEAD POSE TO THE ONE SHOWN IN THE IMAGE ABOVE.
IF NONE IS APPLICABLE, PLEASE CHOOSE N/A.


-90°


-45°

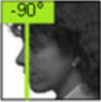

0°



+45°



+90°

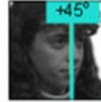
N/A


STEP 2 CLICK ON ONE OF THE HEAD POSE RANGES WHICH THE FACE SHOWN ABOVE IN RED IS MOST LIKELY TO FALL IN.
IF NONE IS APPLICABLE, PLEASE CHOOSE N/A.


-90°



-45°


0°


+45°


+90°

N/A



RANGE 1	RANGE 2	RANGE 3	RANGE 4	RANGE 5	RANGE 6	RANGE 7	RANGE 8	RANGE 9
VERY CLOSE TO -90°	BETWEEN (-90°, -45°)	VERY CLOSE TO -45°	BETWEEN (-45°, 0°)	VERY CLOSE TO 0°	BETWEEN (0°, +45°)	VERY CLOSE TO +45°	BETWEEN (+45°, +90°)	VERY CLOSE TO +90°

Figure 4.5: Example of manual labelling process used for McGill dataset [20]

Typically, these methods use rendering techniques to create ground-truth data (as in [11] [16]), and contain high resolution images of 3D morphable models. Normally a 3D face model is placed on a virtual ground and the camera is moved randomly on a sphere surface whose centre is the same as the head model centre. The images are then obtained by changing the camera view, which is equivalent to rotating the head around the three angles.

The main drawbacks of ground-truth data collected through synthetic models are the following: (I) the face models used are not representative of the real population (age, gender, ethnicity, expression, etc.); (II) the background and also some head parts might be missing from the images. Both these characteristics make it difficult to assess whether the HPE methods would generalize well for more realistic surroundings.

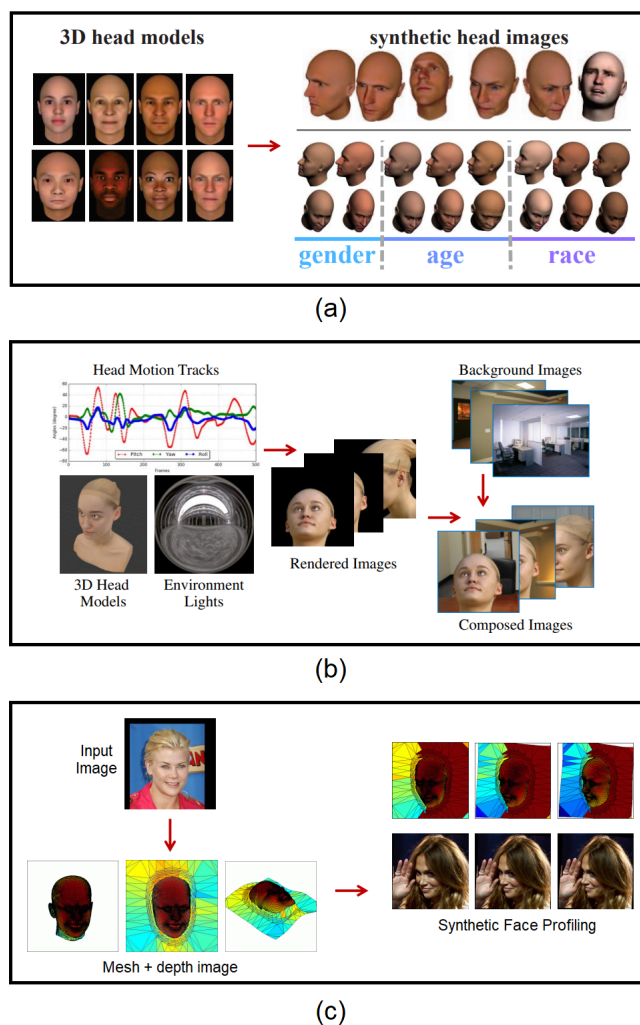


Figure 4.6: Synthetic Head Pose dataset generation: (a) Synthetic [16] used different face models to generate synthetic data, taking into account different attributes; (b) SynHead [11] used a more complex model considering also environment light and background to increment the dataset dimension; (c) 300W-LP [4] model of face profiling to generate new synthetic data (images from [16] [11] [4])

Another method to generate new images is the face profiling adopted to generate new samples in 300W-LP dataset [4]. A 3DMM is fitted to a frontal image to obtain a depth image from 3D mesh, then by rotating it new synthetic images are generated.

- *Directional suggestion*: In a nutshell, ground-truth data are collected by telling each candidate to look into specific marked points in the measurement room, while a fixed camera captures images for each viewing direction.

Such method is comparatively a poor source for creating a ground-truth: first, it assumes that each candidate is accurately directing his/her head towards a specific point, unfortunately, this is a subjective task that people tend to perform rather poorly; furthermore, it assumes that each candidate's head remains in the same accurate physical location, which is not possible [44].

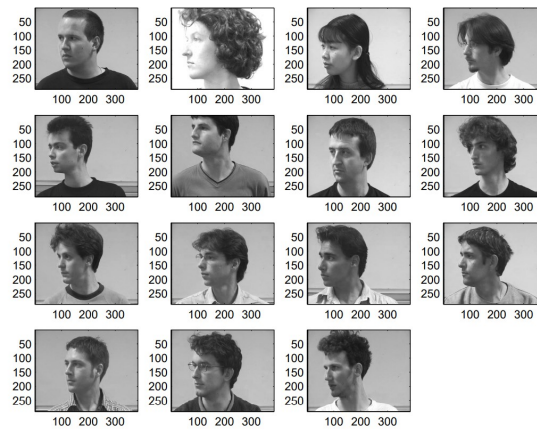


Figure 4.7: shows 15 subjects from Pointing'04 database [29] with head yaw angle 45° and pitch angle 0° . Inconsistencies can be observed between the appearances and the head poses, indeed the appearance of some subjects looks more like a pose of 90° yaw angle. The acquisition method is not effective (image from [44])

- *Laser pointer directional suggestion*: This method is somehow similar to the previous one, with the only difference that a laser pointer is fixed to each subject's head, this allows the subject to pinpoint the discrete locations in the room with much higher accuracy from visual feedback. However, the head is still assumed to remain in the same exact location, which again can introduce errors (people have a tendency to shift their head position during data capture) [73].

- *Camera ring*: Also this approach uses a single camera to capture images of a person and the rotation angle information, but in this case the camera is mounted on an annular ring that rotates around the subject that is seated at the centre. This method has two disadvantages: (I) it assumes that each candidate's head remains in the same accurate physical location during data acquisition; (II) it requires more equipment than other methods.
- *Camera arrays*: In this approach, multiple cameras at known positions simultaneously capture images of a person's face from different angles. If care is taken to ensure that each subject's head is in the same location during capture, this method provides a highly accurate set of ground-truth. Since there is a limitation in the number of cameras, this method cannot be applied to a scenario interested in determining continuous pose ranges.

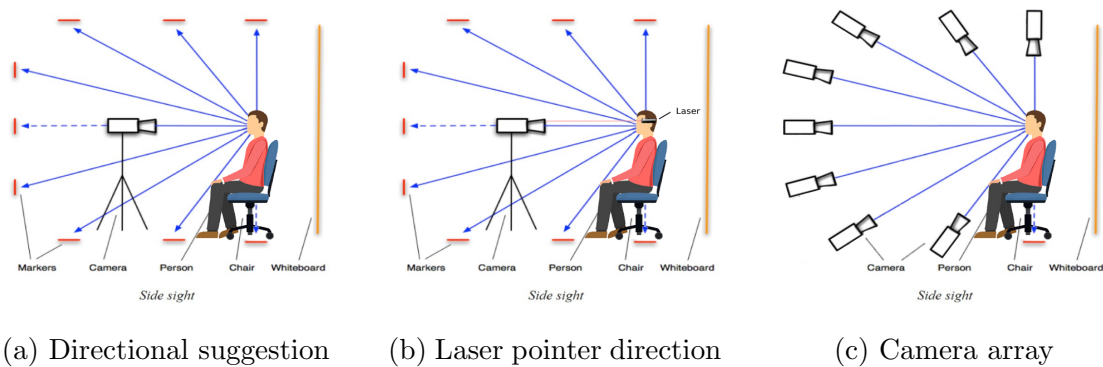


Figure 4.8: Some methods for Head Pose datasets creation

- *Magnetic sensors*: Magnetic Sensors, such as the Polhemus FastTrak or Ascension Flock of Birds, work by emitting and measuring a magnetic field. The sensor can be affixed to a subject's head and used to determine the position and orientation angles of the head [45].

By exploiting a magnetic field to perform the measurements, this method is able to collect a very accurate ground-truth. The main drawback is that these sensors are highly sensitive to the presence of small metals in the environment. For this reason, the circumstances in which data can be collected

are severely restricted and certain applications, such as automotive head pose estimation, are therefore impossible with these sensors.

- *Inertial sensors*: As for metallic and pointer sensors, also inertial sensors are fixed to a person's head [46]. The most commonly used ones are accelerometers and motion-sensing devices, which are normally coupled with Kalman filters for noise reduction. Some less expensive sensors are also available in the market (e.g., Mindflus, InertiaCube), but these do not accurately locate the head. In some cases, these sensors can be included in a pair of wearable glasses, such as the SMI-Eye Tracking glasses² [15]. The main advantage is that, unlike magnetic sensors, these ones are not affected by metallic interference.
- *Optical motion capture system*: These systems are robust and expensive deployments that are, in their most professional form, used for professional cinematic capture of articulated body movement. Typically, an array of calibrated near-infrared cameras use multiview stereo and software algorithms to follow reflective or active markers attached to a person. For head pose estimation, these markers can be affixed to the back of a subject's head [47] and used to track the absolute position and orientation, as shown in figure 4.9.
- *Panoptic studio*: Motion capture technology has come a long way in the past few decades, but the new technology of Panoptic studio has been developed to try to capture motion accurately without those annoying little markers used in optical motion systems [8].

The Panoptic studio is a massively multiview system that utilizes an enormous spherical dome and 480 VGA synchronized cameras, 31 HD cameras, 10 Kinect 2 sensors and 5 DLP projectors, designed to reconstruct the labelled time-varying 3D structure and motion of multiple people engaged in social interaction.

The number of large views provide precision over the captured space and

²<https://imotions.com/hardware/smi-eye-tracking-glasses/>

facilitate the boosting of weak 2D human pose detectors into a strong 3D skeletal tracker.

If we consider only the task of head pose, the main disadvantage of this method is that the cost of implementing this data acquisition system can be very expensive (see figure 4.11).

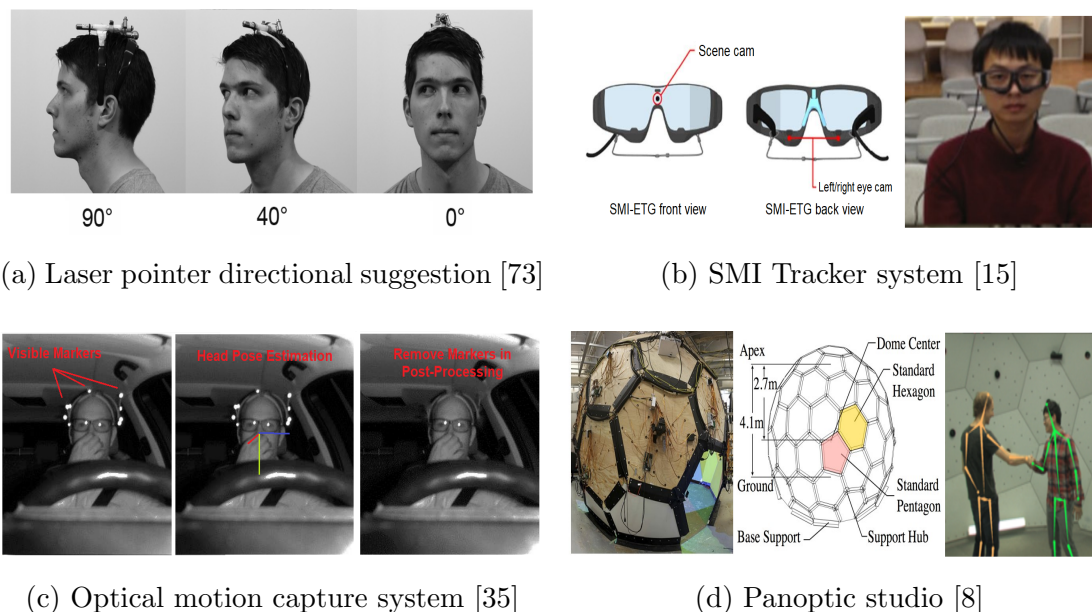


Figure 4.9: Other methods for Head Pose datasets creation

- *Iterative closest point*: Technically speaking, the ICP [43] is an algorithm employed to minimize the difference between two clouds of points. Given 3D data in sensor coordinate system and given a model shape in the model coordinate system, it estimates the optimal rotation and translation that aligns the model shape and the data shape, minimizing the distance between the two.

Therefore, starting from a 3DMM of a face and an RGB-D image, that can be obtained using various tools such as laser scanners, time-of-flight cameras or using regular cameras (Microsoft Kinect is the most used in the literature) and applying stereo vision techniques to the left and right images, the ICP

algorithm [43] can be used to adapt a generic face template to the RGB-D image. Although this method does not provide perfect estimates of the pose, it has been proven that the mean translation and rotation errors were around 1 mm and 1 degree respectively [5] [11].

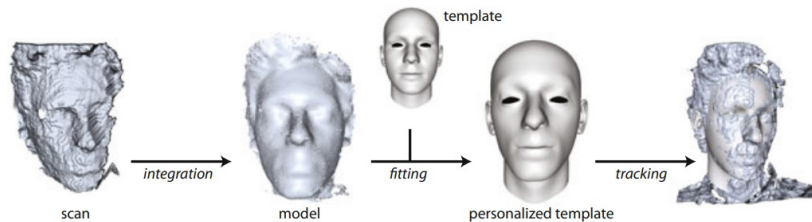


Figure 4.10: Automatic pose labelling using ICP algorithm [43]: A user turns the head in front of the depth sensor, the scans are integrated into a point cloud model and a generic template is fit to it. Personalized templates can be used for accurate rigid tracking (image from [5])

- *Estimation with Neural Networks or other algorithms:* In some cases, information about head pose are computed using deep learning models trained on other HP datasets. The advantage of this approach is that can be obtained data for “in-the-wild” images; the drawback is that the quality of these annotations heavily depends on the model and the training data used, the annotations may be erroneous in many cases.

Other approaches use algorithms that, starting from facial landmarks (typically 21 or 68 landmarks) and a 3D model, try to minimize the distance between the projections of the corresponding points on the 3D model and the actual landmark locations in the image (e.g. POSIT algorithm [23] [75]). The estimated pose is coarse, but nevertheless gives a valid ground-truth for approaches trying to find a rough approximation of the pose.

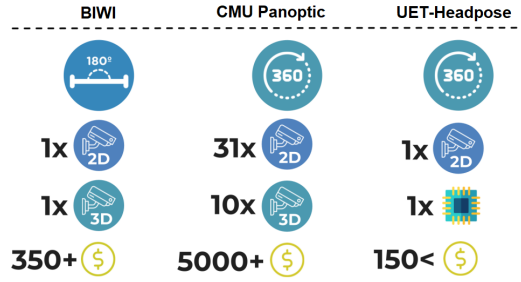


Figure 4.11: Cost comparison of annotations acquisition of different datasets that exploit different methods (image from [2]).

4.4 Evaluation metrics

A common informative metric used for evaluating HPE frameworks is the **Mean Absolute Error** (MAE) for all the three angles, i.e., pitch, yaw, and roll. MAE is quite popular (most of the papers discussed in this thesis use it as main evaluation metric) since it provides a single statistic that gives a quick insight into the performance, for both fine or coarse pose estimations.

$$MAE = \frac{1}{n} \sum_{i=0}^n (|y_i - \hat{y}_i|)$$

However, in scenarios with large-range pose variations (360°), this evaluation method will not be reasonable. For example, when the actual angle is 170° and the predicted angle is -170° , then the two angles are only 20° apart, but the MAE value calculated is 340° , making it bigger than its actual value [2].

For this reason, another measure has been proposed in the literature, called **Mean Absolute Wrapped Error** (MAWE) [61] [2]. The difference is clear if we look at the formula:

$$MAWE = \frac{1}{n} \sum_{i=0}^n \min(|y_i - \hat{y}_i|, 360 - |y_i - \hat{y}_i|)$$

Another measure, mainly used for coarse head pose estimation, is the so-called **Pose Estimation Accuracy** (PEA). Being an accuracy measure, this metric

depends on the number of poses, and therefore gives little information about the actual system performance (was a nearby pose selected, or was the misclassification a widely incorrect estimate?), for this reason few recent researches use it.

Confusion matrices are also employed for the representation in tabular form when classification in several families is performed; row entries are normally indexed with ground-truth and column entries with predicted poses. Also known as error matrix, a confusion matrix provides a quick visual feedback about classification errors, since all correct predictions are located in the diagonal of the table, and sources of errors can be investigated by looking at their spread outside the main diagonal.

In recent studies on head pose estimation in the driving context, new evaluation metrics have been proposed [32] [33] [35]; however, none of the studies on general head pose estimation use them.

The first metric is the **Balanced Mean Angular Error**, introduced to address the problem of the higher number of frontal pose images during evaluation, which leads to an unbalanced amount of different head orientations. The idea is to split the dataset in bins based on the angular difference from the frontal pose and average the MAE of each of the bins [32]

$$BMAE = \frac{d}{k} \sum_i \phi_{i,i+d} \quad i \in d\mathbb{N} \cap [0, k]$$

where $\phi_{i,i+d}$ is the MAE of all hypotheses, the angular difference between the ground-truth and frontal pose is between i and $i + d$, d is the bin size and k is the maximum angle degree considered.

Other two metrics employed are the **Standard Deviation** (Std), which provides insights into the error distribution around the ground-truth, and finally, the **Root Mean Squared Error**, to weight larger errors higher.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2}$$

RMSE takes the squared difference of the predicted value and the ground-truth value, weighing larger errors higher. Thus, high variation in predictions

of an algorithm results in a higher overall error compared to the mean without squaring the values [33].

4.5 Methods

The approaches used in the literature to solve the task of head pose estimation are quite different between them, they have different degrees of automation, different prerequisites and are based on different assumptions.

We try to arrange each system by the approach that underlies its implementation (taking as reference classifications proposed in previous works [55] [58]), by giving a description and evaluating advantages and disadvantages of each approach.

Since head pose estimation has been developed for a long time, many methods have emerged during this period, however, most current work is based on deep learning. Indeed, starting from 2015 the methods based on convolutional neural networks have been used more and more, highlighting a shift in methodology, from traditional machine learning (ML) methods towards deep learning (DL) approaches.

For this reason, we defined the category of classical methods that includes all the approaches that are little, or no longer, considered in the current research and we focused our review on deep learning based models. More details about classical methods can be found in [55] [56]. Other more recent works are [57] [58], with respect to them we will cover the parts relating to the state-of-the-art models in more detail, with a special focus on multi-task learning, 3DMM based and CNN based models.

Classical Methods:

- Appearance template methods: compare a face image to a set of exemplar templates to find the most similar view [45, 102];
- Detector array: use a series of head detectors, each trained for a specific pose and assign the pose relative to the detector with the greatest support [103, 104, 201];

- Manifold embedding: embed an image into low-dimensional manifolds that model the continuous variation in head pose and use these for pose regression [125–134];
- Tracking methods: use temporal constraints to recover the pose from observed movements in video frames [17, 46, 135–138];
- Model based methods: use facial keypoints to determine the head pose from their geometrical configuration using geometric formulation [139–141] or through keypoint matching with a static face model [142–146];
- Non-linear regression methods: use classical non-linear regression machine learning models to develop a functional mapping from the image, or feature data, to a head pose measurement [5, 15, 45, 47];
- Hybrid classical approaches: combine one or more of the aforementioned methods in a single model [55, 56];

Modern approaches:

- Semantic based methods: compute head pose using probability maps produced by a face segmentation algorithm [96–100];
- Model based methods: modern model based approaches use CNNs to regress head pose from landmarks configuration [41, 147–149] or to reconstruct 3DMM and learn its rotation parameters [157, 158, 160, 164].
- Deep learning regression: use deep convolutional neural networks to develop a mapping from the image to the head pose measurements [7, 11, 61, 62, 79, 168, 173–175];
- Multi-task methods: jointly solve head pose with other correlated tasks (e.g. face detection or face alignment) to improve the overall performance [108–113, 116, 119–124];

We tried to organize the different proposed approaches under a unique classification. This has been a quite challenging activity because the borders between

the categories are not well defined, but are blurred, there are methods that fall more into one area, but are also influenced by other approaches. We have taken as a reference the fundamental approach that underlies the implementation of a model to categorize it, e.g. some methods that are based on keypoints, but exploit deep networks for regression, that are classified as model based [147, 148]. The subdivision into two macro categories helped us to show and discuss the evolution of the different techniques up to SOTA methods like: POSEidon [7], WHENet [61], img2pose [60], MNN [119], SynergyNet [160] and SADRNet [164].

A. *Appearance template methods*

Appearance template methods use image-based comparison metrics to match a view of a person’s head to a set of exemplars with corresponding (discrete) pose labels. In the simplest implementation, to an image is given the same pose that is assigned to the most similar of the exemplars [45] (1999) [102] (2002).

Appearance template methods are among the first approaches adopted for the head pose estimation task. Their advantages are (I) the fact that are suitable for both low and high resolution images; (II) the fact that no negative training data is needed during the training process; (III) the fact the expansion of the template models can be easily adjusted at any time, allowing the architecture to adapt to varying conditions, if required.

However, they are limited from the accuracy of head detection models, which must be reliable. In the case of head localization errors, this leads to serious degradation in terms of accuracy. Nevertheless, the greatest weakness lies on the faulty assumption of pairwise similarity. To make this point clearer consider two images of the same subject but in different poses, and two images of different subjects but in the same pose. In this latter case, there is a high probability that the effect of identity causes a wrong association of the image with an incorrect pose. Moreover, these methods become unreliable when variation in local appearance occurs, there is no universal solution to deal with occlusions, such as subjects with eyeglasses, facial hair, etc.

Nowadays there are more efficient approaches that do not suffer from these

limitations, this is why appearance template methods are no longer used.

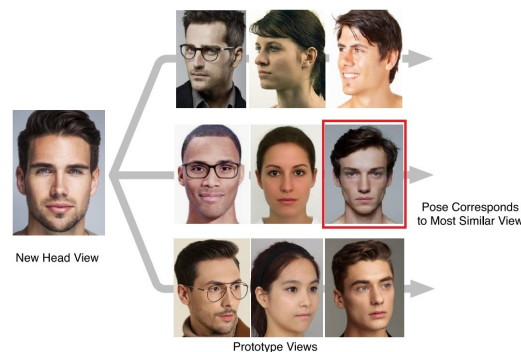


Figure 4.12: Appearance template method: compare a new head view to a set of training examples, each labelled with a discrete pose, and find the most similar view (image from [55])

B. *Detector array*

The idea is to train multiple face detectors, each specific to a different discrete pose, then a test image is evaluated by a sequence of trained detectors, for arrays of binary classifiers, successfully detecting the face will specify the pose of the head, assuming that no two classifiers are in disagreement. For detectors with continuous output, pose can be estimated by the detector with the greatest support [103] (1998) [104] (2006) [201] (2004).

These methods are similar to appearance template algorithms in the sense that they also operate directly on cropped face images, however, instead of comparing an image to a large set of individual templates, the image is evaluated by a detector trained on many images with a supervised learning algorithm.

One of the many advantages of the detector array method is that no separate face detection algorithm is required to develop a complete head pose estimation system. These methods, which work for both low and high resolution images, are also robust to appearance variations, which was the main drawback of appearance template methods.

With respect to disadvantages, training many detectors for each discrete head

pose can be a burdensome task, since both positive (face data) and negative (non-face examples) data should be provided in the training phase for each detector. Moreover, as the number of detectors increases, systematic problems also arise: negative and positive training data for detectors of near poses may be very similar in appearance, making the training difficult and not very effective if many poses are considered.

Nowadays methods that directly regress head pose have an equivalent computational cost, but are more easily trainable and can achieve accurate results also for continuous pose estimation. For this reason, also detector arrays are no longer used in real-world applications.

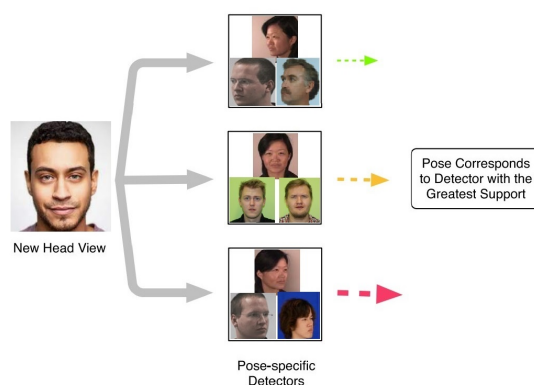


Figure 4.13: Detector array: train a series of head detectors each attuned to a specific pose and assign a discrete pose to the detector with the greatest support (image from [55])

C. *Manifold embedding based methods*

Based on the assumption that informative and discriminative representation of the data lies on a low-dimensional smooth manifold (differentiable manifold), the idea is that the continuous variation of head pose can be modelled and interpreted as a low-dimensional manifold [131]. The low-dimensional representation of the head pose images can be learned by unsupervised or supervised manifold learning.

For head pose estimation, the manifold must be modelled to maintain certain

local attributes (or geodesic distance) of the high-dimensional pose image set, and an embedding technique is required to project a new sample into the manifold. This low-dimensional embedding can then be used for head pose estimation with techniques such as regression in the embedded space or embedded template matching [57].

Any dimensionality reduction algorithm can be considered an attempt at manifold embedding, but the challenge lies in creating an algorithm that successfully recovers head pose while ignoring other sources of image variation [55].

Therefore, these approaches are mainly composed of two stages: a first stage where a feature set is obtained from row image, and a second stage where linear or non-linear methods make use of labelled training set to create a mapping from images (features space) to their corresponding poses [133].

The linear methods have the main advantage that embedding can be easily performed by matrix multiplication. However, the representation ability of non-linear methods is usually superior to the one offered by linear approaches.

Nevertheless, pose data may be located in multiple different (low-dimensional) manifolds, owing to changes in appearance, such as different gender, identity, and lighting. Therefore, some methods based on multi-manifold have been proposed [129] [132].

Early studies tried to estimate head pose by projecting an image into a PCA subspace and comparing the results to a set of embedded templates [125] (1998), obtaining better performances than appearance template methods. The limitation of linear models, like PCA, is that there is no guarantee that the primary components will relate to pose variation rather than to appearance variation. Other approaches focused on non-linear methods, such as Isometric features (Isomap) [126] (2004) [127] (2007), Locally Linear Embedding (LLE) [127] (2007) and Laplacian Eigenmaps (LE) [127] (2007) [131] (2017), the problem is that each of these techniques operates in an unsupervised fashion, ignoring the pose labels, as a result, they have the tendency to build manifolds for identity as well as pose [127] [131].

Sundararajan et al. [130] (2015) address the problem of the source variation by learning a similarity kernel through geometric invariant features using local keypoint correspondence, outperforming previous methods.

An interesting possibility to enhance the embedding results is to adopt a supervised strategy and use head pose labels in order to learn the manifold structure. For example, Balasubramanian and Panchanathan [127] (2007) presented a Biased Manifold Embedding (BME) framework in which the distance metric between features is modified so that heads under similar poses are brought closer to each other than they would be under the unbiased (unsupervised) case. Huang et al. [128] (2011), instead, used supervised Local Subspace Learning to learn a local linear model which showed prominent potential to provide accurate head pose estimation when the training data is pretty sparse and non-uniformly sampled.

Among multi-manifold methods, the integration of manifold embedding and clustering to design an HPE system, which is identity independent, was presented by Liu et al. [129] (2010). They argued that a single manifold is not enough for head pose estimation and that appearance variations, such as changes in identity, scale and illumination, make it necessary the use of multiple different manifolds to model pose parameters. Thus, the authors presented a clustering method, called *K-manifold clustering*, to construct multiple manifolds, each of which characterizes the underlying subspace of some subjects. Peng et al. [132] (2014) also learned multiple manifolds, they used Homeomorphic Manifold Analysis to build a separate manifold for each subject, and learn non-linear mappings to relate each subject-manifold with a common pose-manifold whose topology is predefined as a unit circle or sphere.

More recently, a manifold embedding based on Generalized Discriminative Common Vectors (GDCV) that allows better modelling of a face image subspace was proposed by Diaz-Chito et al. [133] (2018). Finally, Derkach et al. [134] (2019) proposed a 3D head pose estimation algorithm based on non-linear manifold learning using tensor decomposition to generate separate subspaces for each variation factor. They showed that the coefficients within each of these sub-

spaces define a continuous curve that can be modelled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. The proposed model, based on trigonometric functions, produces competitive pose estimation results when RGB-Depth images are available.

Summing up, promising performance is achieved by the classical manifold learning methods, which, however, are highly improved by supervised manifold learning. It proves that the supervised information represented as angles of head poses is helpful in head pose estimation. However, there are still hurdles to take. Most of the methods are tested in different settings, e.g., different databases are used in different methods. A common framework could help to offer fair justifications. Moreover, manifold embedding is still challenging when in-the-wild images are considered given the intrinsic difficulty in modelling the manifolds when more variation factors are present. This is also due to the problem of heterogeneity of training data, since in most cases it is not possible to obtain a regular sampling of different poses for different individuals.

Nowadays other methods are preferred to manifold embedding because they can obtain very accurate real-time performance in easily implementable frameworks. Further research on manifold embedding methods is needed in order to fill the gap with state-of-the-art models.

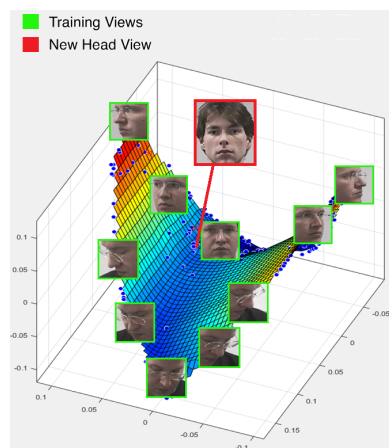


Figure 4.14: Manifold embedding: directly project a processed image onto the head pose manifold using linear and non-linear subspace techniques (image from [131])

D. *Tracking methods*

Tracking methods track the face movement between consecutive video frames and from these infer the pose change by utilizing temporal information of already tracked head parts and *smooth motion constraints* [56].

In the bottom-up approach, the tracking is based on the facial landmarks extracted from each frame (figure 4.15). These feature points are then matched with SIFT descriptors or with 3D face shapes to recover the pose change under full perspective projection [135] (2001) [46] (2003) [136] (2006).

Tracking can alternatively employ a model-based approach, by finding the transformation of a 3D model that best accounts for the observed movement of the head. The model used in tracking can be rigid or non-rigid (deformable), the second one can provide more accurate results. To estimate head pose, one simply needs to find the rotation and translation of the model that best fits each new image-based observation [138] (2000) [137] (2001).

These methods demonstrate high accuracy by discovering the small pose shifts between video frames, but they require initialization from a known head position and, typically, the subject must maintain a frontal pose before the system has begun and must be reinitialized whenever the track is lost [55]. Another drawback of tracking approaches is that they are very accurate only in the short-term, due to tracking error accumulation [56]. Moreover, tracking methods can be trained only on datasets that contain video sequences, reducing the amount of available data (especially in unconstrained scenarios).

These methods were of great interest until five-ten years ago when there were no very accurate methods that could carry out a continuous pose estimation in real-time (Hyperface [108], which was the state-of-the-art at the time, took a few seconds for a prediction) and training deep network was difficult due to computational requirements. Now, methods based on deep learning are able to obtain an accurate estimate of head pose at tens of frames per second without incorporating any time constraint, but by estimating the pose from the single frame each time. For this reason, interest in tracking methods has been lost

in recent literature, although there are hybrid approaches that also incorporate time constraints to increase performance [159] (2018).

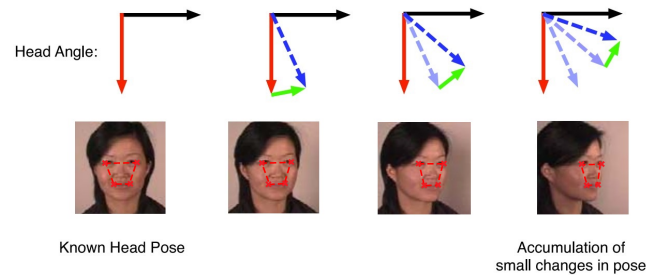


Figure 4.15: Tracking method: track the face movement between consecutive frames and from this infer the pose changes (image from [58])

E. Hybrid classical methods

Hybrid approaches combine one or more of the aforementioned methods to estimate pose. These systems can be designed to overcome the limitations of any one specific head pose category, increasing the estimation accuracy. For example, a static head pose estimation approach can be supplemented with a tracking system. For more details [55] [56]. End-to-end approaches that leverage deep networks are now the most popular systems for head pose estimation, there is no longer the need to combine different approaches to obtain high accuracy.

F. Segmentation based methods

These methods address the problem of head pose estimation by exploiting the strong relationship between the head pose and the position of various face parts. The idea is that the performance of the face pose predictor can be improved if a prior efficiently parsed image, having information about various facial features, is provided as input [96–99].

The first step is to perform *semantic segmentation* over the input image by training a model for each discrete pose previously defined in a specific set. Each model parses the face into different parts (e.g. nose, mouth, eyes, hair) and produces probability maps. Given a new image, the probabilities associated to

face parts by the different pose-specific models are used as the only information for estimating the head pose by using specifically designed algorithms or by training a classifier (e.g. Random Forest, SVMs, etc...).

Huang et al. [101] (2008) were the first to exploit the relation between face segmentation and head pose estimation. In their method, initially, the face is segmented into three face parts (skin, hair, background) and then in a second stage, using a simple regressor, they estimate basic discrete head poses: “frontal”, “right-profile” and “left-profile”. Instead, other works perform a 6-class segmentation and consider a higher number of discrete poses (e.g. 13 poses [96] [98] or 93 poses [97] [99], [100]).

Khan et al. [96] (2017) proposed a simple algorithm to exploit probabilities associated to face parts in order to predict head pose: first, they run segmentation models for all different poses, obtaining probability maps; then, they consider the maximum of such probabilities to assign a pose to each pixel; finally, they count the total number of pixels associated to each discrete pose and assign to the face image that with the highest number. Similarly was done in [97] (2020), but they used the concept of super-pixel, i.e. small meaningful patches belonging to the same object. Instead, other studies used machine learning approaches to obtain the head pose, starting from probability maps multi-class linear SVM classifier [98] (2019), Random Forest classifier [99] (2019) and Soft-Max classifier [100] (2021) have been tested to obtain discrete head poses.

The main advantage of these methods is that are able to exploit the strong relationship between head pose and position of various face parts, which is useful for accurate pose estimation. Moreover, these methods do not require any landmark detection process or face alignment step. Finally, these systems are typically multi-task, they combine HPE, facial expression detection, gender recognition and age classification in a single framework.

However, to build such a systems, manually segmented face images are needed for training, and segmentation is required as preprocessing, therefore the computational cost of such methods can be much higher when compared to other

approaches. Moreover, the proposed methods can solve only the coarse head pose estimation task because they classify images into a discrete set of poses. The use of regression models has not been studied in literature yet.

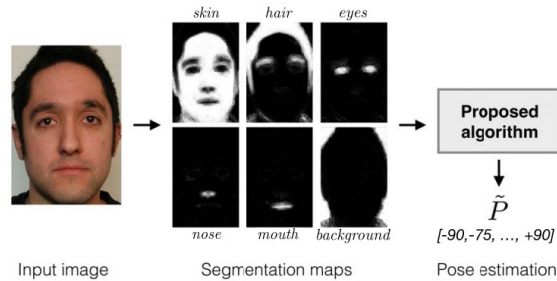


Figure 4.16: Segmentation based method: perform face segmentation and from probability maps infer head pose (image from [96])

G. Model based methods

Model based methods require either a 3D head model or the localization of *facial keypoints* (landmarks), such as eyes, eyebrows, nose, lips, etc. (or both of them in some cases) and from these estimate the head pose. It is proven that these factors, such as the location of the face in relation to the contour of the head, strongly influence the human perception of the head [55]. For this reason, model based methods are particularly interesting, they can directly exploit properties which are known to influence human head pose estimation.

Different facial features and keypoints are exploited in different ways in the literature.

Early approaches, which we can call *Geometrical methods*, focused on a set of facial landmarks and estimate the pose directly from the configuration of these points by using geometric concepts. For example, by computing symmetry axis connected the eyes/mouth and assuming a fixed ratio between keypoints and measuring the deviation from one pose to another, or by using the incident angles between different axes and perspective distortion [139] (1996) [140] (2006) [141] (2010). Geometrical approaches were among the first used to solve the head pose estimation task because require few calculations and are quite simple [56].

However, they have been progressively abandoned with the introduction of non-rigid face models and neural networks that obtained higher accuracy.

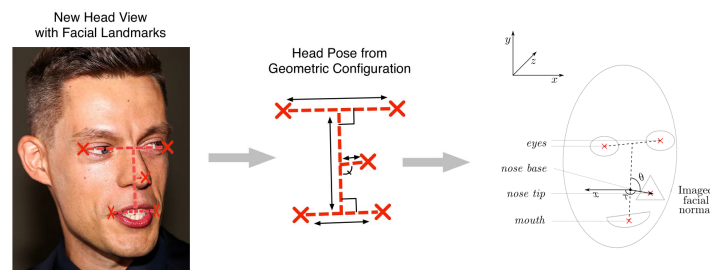


Figure 4.17: Geometrical method: detect facial keypoints and estimate the pose from the relative configuration of these features (image source [55], [141])

In recent years, with the development of deep learning and due to the high availability of data, methods that directly extract facial landmarks have improved enormously their performance and have become the dominant approach in facial analysis tasks [62]. A by-product of face alignment is the ability to recover the 3D pose of the head in two different ways: (I) the *Landmark-to-Pose* approach and (II) by exploiting *deformable methods*.

In the *landmark-to-pose* approach the keypoints are given as input to a ML, or DL, algorithm that regresses the head rotation angles.

Werner et al. [41] (2017) proposed a benchmark protocol to learn pose estimator on top of any landmark detector, called HPFL, that trains a Support Vector Regression (SVR) model using landmarks as features. Gupta et al. [147] (2019) proposed to use a deep learning architecture to regress head-pose giving as input uncertainty maps computed from 5 facial keypoints. Even Xia et al. [148] (2019) used a CNN, but they give as input a *heatmap* of 68 landmarks stacked with a transformed version of the input image, so that the neural network can focus on the area around facial landmarks while extracting features from the image, reducing interference from wild environment. Dapogny et al. [149] (2020) proposed an attentional cascade model that iteratively refines head pose and landmark estimates. The advantage is that using head pose information to refine landmark

alignment provides more precise landmark estimates (as stated also in [154]), which in turn helps refine the head pose prediction, further advocating for an entwined landmark alignment and head pose prediction scheme.

Recently, other researchers have tried to define methods that does not need training for estimating head pose once facial landmarks are detected. Abate et al. [150] (2019) used a quad-tree, i.e. a particular kind of unbalanced tree, that divides the image into smaller and smaller quadrants, to measure the distance between the representation of the input face with a reference model. Barra et al. [151] (2020) exploit a spider-web shaped model that uses the landmark locations to build a feature vector, which in turn is compared to a set of prototypical vectors to determine the closest one and establish the pose. Unfortunately with these two methods only discrete pose can be obtained (with 5° of angular step), they are computationally efficient but less effective than other methods.

Deformable methods, instead, use a non-rigid model and fit it to the image such that it conforms to the facial structure of each individual and estimate the head poses from the correspondence between feature points on a 2D face image and those on a 3D facial model.

The 3D pose information of the head can be inferred by solving the *Perspective-n-Point* (PnP) problem, i.e. the problem of estimating the pose of an object by finding the rotation matrix R and the translation vector t given intrinsic camera parameters, known locations of n 3D points and their corresponding 2D projection in the image. Indeed, by looking for the projection relation between a 3D facial model and a 2D face image, head pose angles can be calculated from the elements in the rotation matrix directly [142] [143].

The most simple and commonly used pipeline involves a number of steps [62]: (1) face alignment; (2) definition of 3D human mean face model; (3) approximation of camera intrinsic parameters; (4) solving 2D-3D correspondence problem using one of the available PnP algorithms, such as POSIT [75] or DLS [76]. These methods became very popular, and replaced geometric methods, because there is no need to include and train a pose estimation model, any method for face

alignment can be used, Dlib [152] (2014) and FAN [153] (2017) obtained accurate results in HPE (a survey on face alignment methods [77]).

Other works do not use a mean face model, but try to model precisely the facial structure. Among them, early deformable approaches [155] (2006) [156] (2012) were based on Active Shape Models (ASM) [64] and on Active Appearance Models (AAM) [65], these are statistical shape models made out of object samples. ASM and AAM guide the learning algorithm to iteratively deforms the model to find the best match position between the model itself and the data in a new image [55]. The fitting procedure is very fast, but one major drawback is its non-robustness to viewpoint changes. In the case of facial images, ASM/AAM fitting is not appropriate for adopting to faces that exhibit large pose variations. For multi pose fitting a combination of a small number of 2D AAM models is needed [167] (2009).

To overcome the drawbacks of the 2D AAM, modern deformable approaches rely on 3D Morphable Models (3DMM). A 3DMM can be fit to image data or depth or shape data to adapt the model to the subject's head, covering larger pose variations, then the 2D-3D correspondence can be solved more efficiently.

Wu et al. [157] (2017) assumed to have a 3D deformable facial model and followed a cascade iterative procedure that iteratively updates the facial landmark locations, the head pose angles and non-rigid deformations. There is no learning involved for head pose that is estimated from the 3D deformable model by minimizing the projection error for all landmark points. Diaz Barros et al. [159] (2018) proposed a hybrid method that incorporates two strategies: (1) a temporal tracking scheme, which uses optical flow to compute the correspondences of a set of keypoints in every pair of frames; (2) a head pose estimation scheme which estimates pose independently in each frame by aligning 2D facial landmarks to every image; the head pose in each scheme is estimated by minimizing the re-projection error from the 3D-2D correspondences. Liu et al. [158] (2021) trained a CNN to reconstruct a personalized 3D face model from the input head image and through 3D-2D keypoints matching estimate head pose under constraint perspective transformation (see figure 4.18).

Furthermore, some of the state-of-the-art networks for head pose estimation follow a different approach, also based on 3DMM. They focus on the *3DMM-based 3D dense alignment 3D dense reconstruction* task and can be also used for pose estimation, indeed, 3DMM regression contains pose, shape and expression parameters. There is no keypoints matching involved.

Zhu et al. [4] (2016) proposed an alignment framework termed 3D Dense Face Alignment (3DDFA), which directly fits a 3D face model to RGB images via convolutional neural networks. The primary task of 3DDFA is to align facial landmarks, even for the occluded ones, using a dense 3D model. As a result of their 3D fitting process, the 3D head pose is produced. SynergyNet [160] (2021) is a novel network designed to predict complete 3D facial geometry, including 3D alignment, face orientation and 3D face modelling. The network defines a synergy process that utilizes the relation between 3D landmarks and 3DMM parameters to improve the overall performance. Despite the large amount of work on 3DMM-based 3D dense alignment and the fact that many of the proposed approaches directly estimate rotation matrices, Wu et al. were the first to propose a discussion on the head pose estimation task, previous works only focus on the evaluation of landmarks and 3D faces. The authors, as well as evaluate SynergyNet, conducted extensive and detailed benchmarking on other 3DMM-based methods, such as 3DDFA-TAPAMI [161] (2017), 2DASL [162] (2020) and 3DDFA-V2 [163] (2020), highlighting the better performance of the proposed network due to the innovative synergy process introduced. SADRNet is another network proposed very recently by Ruan et al. [164] (2021) that is one of the state-of-the-art models on AFLW2000 [4] dataset. This is an encoder-decoder-based architecture that regresses the deformation D and infers the pose parameters f , R and t to reconstruct the 3D face geometry from a single 2D face image. The most important novelty introduced in the network is the attention mechanism used to enhance the visible facial information and estimate the transformation matrix only with visible landmarks, giving robustness to occlusions and large pose variations.

Finally, with the development of consumer-level depth-image sensors, many

studies have tried to exploit 3D-face model-based approaches using RGB-D data. These studies have developed in parallel with the others presented before and mainly use *optimization techniques*, such as the ICP algorithm [43], which aim to minimize the discrepancy between depth data and a parametrized 3D model. Martin et al. [165] (2014) proposed a real-time head pose estimation method that first creates a point-cloud based 3D head model from the input depth image and then registers the 3D head model with the iterative closest point (ICP) algorithm [43] for head pose estimation. Mayer et al. [166] (2015) proposed estimating head poses by registering a 3D morphable model (3DMM) to the input depth data through a combination of particle swarm optimization (PSO) and the ICP algorithm [43]. Higher pose estimation accuracy is achieved at the expense of a much higher computational cost. A 3D morphable model and online 3D reconstruction are used by Yu et al. [10] (2018) for full head pose estimation, thus also handling extreme poses. Although estimating the head poses on the depth image can avoid suffering from the cluttered background and illumination changes, that are common in RGB images, the main disadvantage is that depth image sensors are not available in most of the current real-world applications.

Summing up, we saw that there is a huge literature of approaches based on the facial keypoints, that are used as key elements of geometric methods, or given as input to neural networks (so used as features), or even are the only information needed in the PnP approach. The advantage is that it has been demonstrated that there is a close relationship between head pose and the distribution of the landmarks, these are valuable information to estimate head pose [148]. Moreover, there is a growing number of landmark detectors/trackers that can be used for research purposes for free and there is a rapid progress in improving the landmark quality, including unconstrained scenarios with difficult lighting, out-of-plane head poses, and occlusions [41].

PnP approach is one of the most used in the literature, but has a disadvantage, many parameters (such as camera pose) typically are approximated and this can lead to inaccuracies in the results. Moreover, when a mean face model is used, even with perfect registration, the images of two different people will

not line up exactly, since the location of facial features varies between people, leading to errors in the final result [148]. For this reason, recently developed approaches rely on face reconstruction as previous step to 2D-3D keypoints matching [158]. Furthermore, these methods need high-resolution images and the position of landmarks must be initialized before the pose estimation, as discussed by Mallick [142].

For this reason, recent research is mainly focused on landmark-to-pose approaches that regress the head pose from landmark configuration using deep networks, and on 3DMM based approaches that reconstruct and align a 3D dense face model with the images. Less research has been done in this last case, but this seems a very interesting approach that obtains impressive results, even if the head pose is obtained only as a by-product. The disadvantage of 3DDFA approaches is that the networks are quite complex, fully supervised and depends on costly face mesh annotations. Nevertheless, SADRNet [164] reconstructs the 3D model of the face (starting from a cropped image) in 13.5 ms. Although, there is a lot of uncertainty about the performance in low resolution far-field imagery due to the difficulty in achieving good fitting and precise image feature location in those conditions.

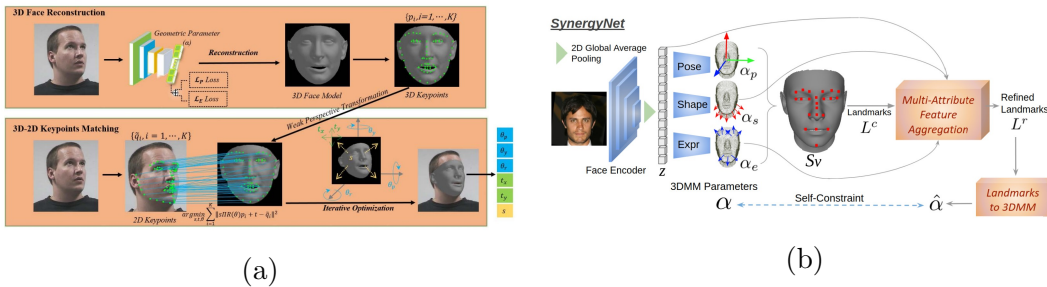


Figure 4.18: Example of models based on 3D face reconstruction: (a) A personalized 3D face model is reconstructed from the input head image using a CNN, then keypoints matching is used to obtain the pose [158]; (b) In SynergyNet a backbone network learns to regress 3DMM parameters (pose, shape, expression) [160].

H. *Non-linear regression methods*

The non-linear regression methods do not require keypoints detection, but directly predict the head pose angles through images. A model is trained in a supervised manner and learns a functional mapping from the image space to discrete/continuous pose directions. The main challenge is to train a model in a way to ensure that the regression tool will learn a proper mapping.

Early approaches used classical machine learning models such as Support Vector Regressor (SVR) [45] (2000), Localized Gradient Histograms (LCH) [47] (2007) or Random Forest (RF) [5] (2011) [15] (2016). Then, widely used became feed-forward neural networks: Multi-layer Perceptron (MLP), with cropped images of the head, and Locally Linear Maps (LLM) obtained good results in the pose estimation task [55, 58].

In the last years, there was a shift towards the deep learning paradigm, with the increasing use of convolutional neural networks to estimate the three-dimensional head pose from image intensity with higher accuracy.

First attempts with deep models exploited simple architectures [169] (2014) [170] (2016) and common networks [172] (2017), such as AlexNet [71], VGG [70], ResNet [69]. Patacchiola et al. [171] (2017) improved the results by introducing dropout and adaptive gradient methods during the training of the network, and by training a different specialized network for each rotation angle (pitch, yaw, roll), that permits fine-tuning for a specific degree of freedom without losing predictive power on another one. They also released an open-source Python library called DeepGaze³. Work from Gu et al. [11] (2017) used a recurrent neural network to regress the head pose Euler angles by exploiting the time dimension in video sequences. RNN has the ability to learn motion information implicitly, gaining robustness to large head pose variations and occlusions.

Ruiz et al. [62] (2018) proposed to use a three-branch convolutional neural network structure, that they called Hopenet, where each branch is responsible for one of the Euler angles. All branches share a backbone network that can be of

³<https://github.com/mpatacchiola/deepgaze>

arbitrary structure, e.g. ResNet50 [69], AlexNet [71], VGG [70]. This backbone network is augmented with a branch-specific fully-connected layer that predicts a specific angle. By having three cross-entropy losses, one for each Euler angle, three signals are backpropagated into the network, which improves learning.

The overall framework of Hopenet is adopted also by Zhou et al. [61] (2020) for their network WHENet. WHENet adopted a lighter backbone w.r.t. previous work, EfficientNet-B0 [72] was used (it incorporates Inverted Residual Blocks, from MobileNetV2, to reduce the number of parameters while adding skip connections). This network is optimized for the full range Euler angles (360 degrees), not only for narrow range as the previous works (180 degrees). This is achieved by careful choice of the wrapped loss function as well as by developing an automated labelling method for the CMU Panoptic dataset [8], that is used during the training of the network.

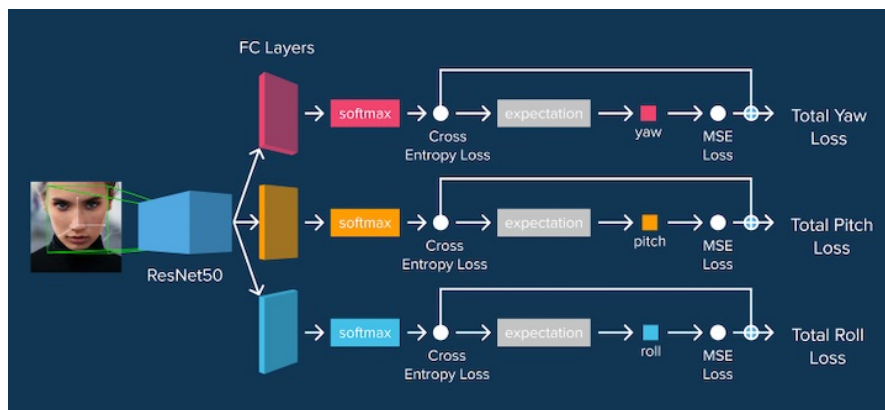


Figure 4.19: Hopenet architecture [62]: ResNet50 with combined Mean Squared Error and Cross Entropy Losses (image from [67])

FSA-Net [168] (2019) introduced a feature aggregation method to improve pose estimation. QuatNet [173] (2019) proposed a Quaternion-based face pose regression framework which claims to be more effective than Euler angle-based methods. TriNet [79] (2021) used a three vector-based representation that replaces Euler-based and Quaternion-based representations for increasing efficacy. RankPose [174] (2020) is another CNN that explored Siamese architecture and

ranking loss to distinguish pose-related from a mixture of pose-related and irrelevant features, such as age, lighting and identity.

Given the fact that the bounding box significantly affects the quality of the trained NN for the HPE problem [187] [199], Sheka et al. [175] (2021) proposed to average the results of predictions of the same neural network, but with various bbox offsets, in what they call *offset ensemble*.

Recently, some attempts to propose lightweight networks that obtain good results at lower costs have been made, Berral-Soler et al. [179] (2021) and Dhingra [180] (2022) proposed respectively RealHePoNet and LwPosr networks. However, the results are less accurate than those obtained with more complex models.

Other researchers, to overcome the limitations of publicly available datasets, that are limited in size, resolution, annotation accuracy and diversity, used synthetic generated data from high-quality 3D facial models to train their networks [16] (2016) [11] (2017). Wang et al. [176] (2019) proposed a coarse-to-fine network to predict head pose trained on synthetically rendered faces. However, they noticed that the difference (domain gap) between rendered (source domain) and real-world (target domain) images negatively affects the performance. For this reason in [177] (2019) and [178] (2021) Domain Adaptation (DA) techniques are applied to reduce the influence of domain differences.

Finally, some researchers leveraged depth data [5] (2011), [182] (2015), [7] (2017). Among them the best performing is POSEidon [7] (2017), which is a network composed of three independent convolutional nets followed by a fusion layer, specially conceived for understanding the pose by depth. This is the state-of-the-art model on the BIWI database [5] (see table 4.3).

The main advantage of head pose estimation derived from CNNs is the strong learning ability, especially for image processing, which make it possible to achieve the desired effects. These algorithms work properly with high and low resolution images, and they have demonstrated their representational ability in tolerating some errors in the training set data. They are not dependent on the head model

chosen, the landmark detection method, the subset of points used for alignment of the head model or the optimization method used for aligning 2D to 3D points. Moreover, they can be computationally efficient, straightforward to implement and easily updated with the addition of new data (data-driven approach, the upper limit is high).

However, the performance of these methods drops drastically if the labelled face images are not properly annotated. There can be difficulties in obtaining sufficient data with head annotations for head pose estimation training, especially data with changes in appearance (such as sex, age group, and race attribute) or environmental interference (such as lighting conditions, shooting angle). Many datasets don't have a uniform distribution of data (many images contain frontal or near-frontal faces) causing difficulties in learning large pose variations. Moreover, most powerful CNNs are complex, the number of layers is getting deeper and deeper, and can require a long training time. All these methods rely on a face detection step, prior to pose estimation, that can heavily influence the result.

We noticed that the problem of the amount of data can be addressed by using synthetic datasets, these provide accurate ground-truth for each pose and contain a uniform distribution of head angles. The disadvantage of synthetic data is the domain gap with real ones, the performance is lower than using real data; further research is needed to overcome this limitation.

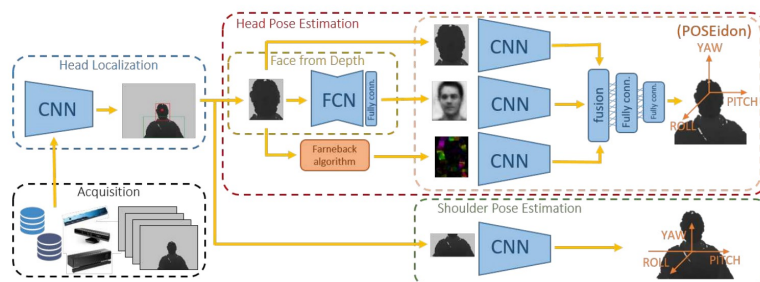


Figure 4.20: POSEidon architecture [7]: depth images are provided to a head localization CNN, then the head region is given in input to the POSEidon network to obtain pitch, yaw and roll estimations (image from [7])

I. *Multi-task methods*

The idea behind multi-task methods is to relate head pose estimation to other face image analysis problems, such as gender recognition, landmark detection, face expression recognition, race classification, etc. because it is proven that jointly solving multiple tasks can lead to better performance [105–114].

The *multi-task learning* (MLT) paradigm encompasses a set of learning techniques that provide effective mechanisms for sharing information among multiple tasks. It enables the use of larger and more diverse datasets, that improve the regularization during training and the generalization of the final model (for more details section 2.5).

Among multi-task methods adopting traditional machine learning frameworks there are [105] (2013) [106] (2014). The former adopts the graph guided FEGA-MTL framework for head pose classification of mobile targets based on multi-view image source. The physical space is divided into a discrete number of planar regions and the model try to learn the pose appearance relationship in each region. The latter tried to do the same, but evaluating the SVM-MTL framework.

However, multi-task methods have become very popular with the explosion of deep learning because of the unique ability of neural networks to transfer and share knowledge among various tasks. MTL has been widely used to simultaneously learn related tasks, such as: face detection + head pose estimation [115] [116] [123] [124] [60], face alignment + head pose estimation [110] [111] [119] [120] [121], face detection + face alignment + head pose estimation [112] [113] [122], face detection + face alignment + head pose estimation + gender recognition [108] [117], or also in combination with other tasks such as face recognition and appearance attributes estimation (age, smile, etc.) [107] [109] and finally there is head pose estimation + gaze estimation [118].

Zhang et al. [107] (2014) were the first to investigate the possibility of optimizing multiple tasks by using a Task-Constrained Deep Convolutional Neural Network (TCDCN) to jointly optimize facial landmark detection with a set of related

tasks, such as head pose estimation. The proposed network learns a shared feature space that is optimized to solve all the tasks at the same time. The network does not perform face detection, therefore it requires an image of a face as input or an additional preprocessing step. A similar network was proposed also by Ahn et al. [115] (2018), but their focus was on real-time driving face detection and head pose estimation.

Ranjan et al. [108] (2017) proposed a new model called Hyperface that performs face detection, face alignment, pose estimation and gender recognition. The network is designed to exploit the fact that information contained in features is hierarchically distributed throughout the network, therefore lower layers respond to edges and corners, and hence contain better localization properties (are more suitable for face alignment and pose estimation tasks); on the other hand, higher layers are class-specific and suitable for learning complex tasks such as face detection and gender recognition. They make use of all intermediate layer features (called *hyperfeatures*) through a technique named *feature fusion*, which allows to transform features to a common subspace where these can be combined linearly or non-linearly. They show that fusing intermediate layers improves the performance for structure dependent tasks of pose estimation and landmarks localization, as the features become invariant to geometry in deeper layers of CNN. Recently, Zhang et al. [117] (2020) revised the Hyperface model by using the differential private stochastic gradient descent to preserve the privacy of the training data during model training.

Then, Ranjan et al. [109] (2017) proposed another model called All-in-One. It differs from Hyperface because (I) simultaneously performs a higher number of tasks and (II) domain-based regularization is adopted by training on multiple datasets, each one specific to a subset of the tasks.

Xu et al. [110] (2017) have brought into the field a new type of network, i.e. a cascaded architecture that is designed in a hierarchical way based on coarse-to-fine principles, which refines the shape and pose sequentially. Other cascaded architectures have been presented in the literature, the main difference among them is the number of stages, the type and the number of tasks addressed in

each stage [113] (2018) [116] (2018).

Kumar et al. [111] (2017) transformed the cascaded regression formulation into an iterative scheme, by proposing the KEPLER model. In each iteration, a regressor predicts visibility, pose and the corrections for the next stage, and a rendering module uses these corrections to prepare new rendered data employed in the next iteration. The network is trained on three tasks namely, pose, visibilities and bounded error using ground-truth annotations. The joint training is helpful since it models the inherent relationship between the visible number of points, the pose and the amount of correction needed for a keypoint in a particular pose.

Many other researchers focused on improving the time needed for the network to resolve the tasks, indeed this is the main drawback of some of the presented models (e.g. Hyperface [108] or All-in-One [109]) that limits real-world applications. Cheng et al. [112] (2018) proposed a model that exploits single-shot object detection module (SSD) to perform multi-scale face detection, face alignment and head pose estimation at the same time at a much higher speed. ASMNet [121] (2021) is a lightweight CNN assisted by an Active Shape Model (ASM) [64], used to guide the network towards learning, that achieved an acceptable performance for face alignment and pose estimation while having a significantly smaller number of parameters and floating point-operations. ATPN [120] (2021) and MOS [122] (2021) focused on defining a network structure with an even smaller number of parameters to augment efficiency. Other architectures, such as Multitask-net [123] (2021) and TRFH [124] (2021), leveraged the feature pyramid network to detect faces on different scales.

Valle et al. [119] (2020) proposed another type of architecture, an encoder-decoder CNN (see figure 4.21). They locate the head pose estimation task at the end of the encoder network, in this way the network bottleneck acts as embedding representing face pose. Instead, visibility and face alignment tasks are located at the end of the decoder, since they require information about the spatial location of landmarks in the image. This is the only paper to propose an encoder-decoder architecture. The presented model, called MNN, achieves

results comparable to the state-of-the-art methods for the head pose estimation task; this is due to the network architecture and to a new training strategy that uses reannotated datasets.

The main advantage of multi-task approach is that many tasks can be solved with a single model. Furthermore, all these tasks are strictly related, therefore the overall performance is improved due to the network's ability to learn correlations between data from different distributions in an effective way, so more discriminative features are learned. Also, some methods perform face detection with head pose estimation, reducing the time needed to perform preprocessing of the image. Another advantage is that multiple datasets can be used for training, increasing the amount of available data.

The main disadvantage of multi-task approach is the lack of public benchmark datasets with all the annotations for all the tasks. It's difficult to compare multi-task models among them and to other head pose estimation methods because they use a different combination of datasets for training and testing, therefore the better performance of a model could be due mainly to the training strategy rather than to the architecture of the proposed network. Moreover, some of the older models were not suited for real-world usage, e.g. Hyperface and All-in-One architectures took 3.5 seconds to process a single image [109]. Although newer models have managed to limit this problem, making it possible to obtain real-time systems.

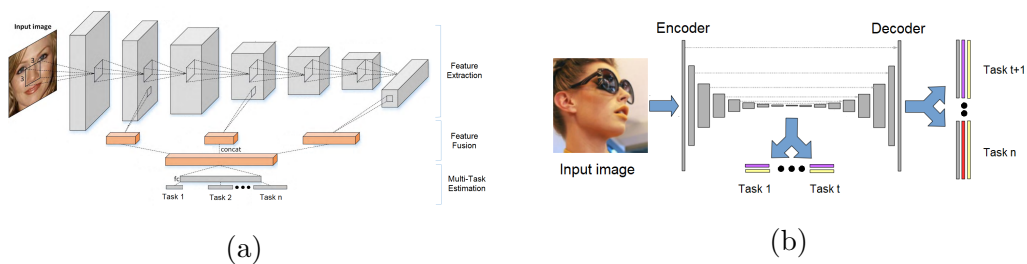


Figure 4.21: Multi-task methods: (a) A convolutional neural network with *feature fusion*, examples are Hyperface [108] and All-in-One [109] (image from [116]); (b) Encoder-decoder network, called MNN, adopted in [119]

Year	Paper	Approach	DoF	Dataset
2011	Fanelli et al. [5]	Random Forest	3	BIWI
2012	Baltrusaitis et al. [21]	CLM-Z Model based	3	BIWI, BU, ICT-3DHP
2014	Ahn et al. [169]	DCNN	3	BIWI
2014	Martin et al. [165]	Model based	3	BIWI
2014	Peng et al. [132]	Manifold embedding	3	Multi-Pie
2014	Tulyakov et al. [17]	ML + Tracking	2	Dali3DHP
2014	Zhang et al. [107]	Multi-task DCNN	3	AFLW ^{***} , AFW ^{***}
2015	Drouard et al. [183]	Gaussian locally-linear mapping	3	BIWI, Pointing'04
2015	Meyer et al. [166]	3DMM Model based	3	BIWI, ETH
2015	Papazov et al. [181]	3DMM Model based	3	BIWI, Synthetic data
2015	Saeed et al. [182]	ML: HoG + SVR	3	BIWI, ICT-3DHP
2015	Sundararajan et al. [130]	Manifold embedding	3	AFLW, AFW, McGill
2016	Gu et al. [11]	RNN	3	BIWI, ETH, SynHead
2016	Liu et al. [16]	DCNN	3	BIWI, Synthetic
2016	Xingyu et al. [170]	DCNN (VGG)	3	IDIAP-HP
2017	Amador et al. [172]	DCNN	3	300W, AFLW, AFW
2017	Barros et al. [144]	PnP Model based	3	BU
2017	Borghi et al. [7]	DCNN	3	BIWI, ICT-3DHP, Pandora
2017	Bulat et al. [153]	PnP Model based	3	300-VW [◊] , 300W-LP [◊] , AFLW2000 [◊] , Menpo [◊]
2017	Diaz-Chito et al. [133]	Manifold embedding	3	CAS-PEAL, CMU-Pie, DrivFace, Pointing'04, Taiwan RoboticsLab
2017	Gao et al. [194]	Deep label distribution learning	3	AFLW, BJUT-3D, Pointing'04
2017	Gou et al. [184]	Model based	3	300W [◊] , BU*
2017	Khan et al. [96]	Segmentation based	2	Pointing'04
2017	Kumar et al. [111]	Multi-task DCNN	3	AFLW ^{*◊} , AFW ^{*◊}
2017	Lathuliere et al. [187]	DCNN	3	BIWI
2017	Patacchiola et al. [171]	DCNN	3	AFLW, AFW, Pointing'04
2017	Ranjan et al. [108]	Multi-task DCNN	3	AFLW ^{†*} , AFW ^{†*◊} , CelebA*, FDDB [†] , LFWA*, Pascal [†]
2017	Ranjan et al. [109]	Multi-task DCNN	3	Adience [□] , AFLW ^{†*◊} , AFW ^{†*} , CASIA [△] , Chalern LAP2015 [□] , CelebA*, FDDB [†] , FG-NET [□] , IJB-A [△] , Morph [□] , Pascal [†]
2017	Wu et al. [157]	Model based	3	BU4D-FE [◊] , BU*, COFW [◊] , Multi-Pie ^{◊*}
2017	Xu et al. [110]	Multi-task DCNN	3	300W ^{*◊}
2017	Yu et al. [185]	Model based	3	BIWI, UbiPose
2018	Ahn et al. [115]	Multi-task DCNN	3	AFLW ^{†*} , BIWI ^{†*} , RCVFace ^{†*} , NDS [†]
2018	Barros et al. [159]	Model based + Tracking	3	BU
2018	Cai et al. [113]	Multi-task DCNN	3	300W ^{†*◊}
2018	Chen et al. [112]	Multi-task DCNN	3	AFLW ^{†*◊} , AFW ^{*◊} , FDDB [†] , Pascal [†] , WIDER [†]
2018	Gupta et al. [147]	Model based MLP	3	AFLW, BIWI
2018	Hong et al. [114]	Multi-task Multi-view + Manifold learning	3	BIWI, Pointing'04
2018	Ruiz et al. [62]	DCNN	3	300W-LP, AFLW, AFLW2000, BIWI
2018	Yu et al. [10]	Model based 3DMM	3	BIWI, UbiPose

2018	Zhang et al. [195]	Multi-task DCNN	3	AFLW [◊]
2019	Abate et al. [150]	Model based Quad Tree	3	AFLW, BIWI
2019	Benini et al. [98]	Segmentation based SVM	2	Pointing'04
2019	Derkach et al. [134]	Manifold embedding	3	BIWI, SASE
2019	Hsu et al. [173]	DCNN	3	300W-LP, AFLW, AFLW2000, AFW, BIWI
2019	Khan et al. [97]	Segmentation based	3	AFLW, BU, ICT-3DHP, Pointing'04
2019	Khan et al. [99]	Segmentation based Random Forest	3	AFLW, BU, ICT-3DHP, Pointing'04
2019	Kuhnke et al. [177]	DCNN	3	Biwi+, SynBIWI+, SynHead++
2019	Liu et al. [196]	DCNN	3	300W-LP, AFLW, AFLW2000, AFW, BIWI
2019	Shao et al. [197]	DCNN	3	300W-LP, AFLW2000, BIWI
2019	Wang et al. [176]	DCNN	3	BIWI, BU, Pointing'04, Synthetic data
2019	Wang et al. [189]	DCNN	3	300W-LP, AFLW, AFLW2000, BIWI
2019	Xu et al. [198]	DCNN	3	CAS-PEAL, Multi-Pie, Pointing'04
2019	Xia et al. [148]	Model based DCNN	3	300W-LP, AFLW2000, BIWI, CAS-PEAL, DriveFace
2019	Yang et al. [168]	DCNN	3	300W-LP, AFLW2000, BIWI
2020	Barra et al. [151]	Model based	3	AFLW, BIWI, Pointing'04
2020	Cao et al. [79]	DCNN	3	300W-LP, AFLW2000, BIWI
2020	Dai et al. [174]	DCNN	3	300W-LP, AFLW2000, BIWI
2020	Dapongy et al. [149]	Model based	3	300W, 300W-LP, AFLW2000, CelebA, WFLW
2020	Ewaisha et al. [118]	Multi-task DCNN	3	CAVE
2020	Valle et al. [119]	Multi-task DCNN	3	300W-LP ^{◊*} , AFLW ^{◊*} , AFLW2000*, BIWI*, COFW [◊] , WFLW ^{◊*}
2020	Wang et al. [143]	PnP Model based	3	300W, AFLW2000
2020	Zhang et al. [188]	DCNN	3	300W-LP, AFLW2000, BIWI
2020	Zhang et al. [117]	Multi-task DCNN	3	AFLW ^{†*}
2020	Zhou et al. [61]	DCNN	3	300W-LP, AFLW2000, BIWI, CMU Panoptic
2021	Albiero et al. [60]	Multi-task DCNN	3	300W-LP*, AFLW2000*, BIWI*, WIDER ^{†*}
2021	Basak et al. [178]	DCNN	3	BIWI, SASE, Synthetic data
2021	Berg et al. [191]	DCNN	3	BIWI
2021	Berral-Soler et al. [179]	DCNN	3	AFLW, Pointing'04
2021	Fard et al. [121]	Multi-task DCNN + ASM	3	300W ^{†*} , WFLW ^{†*}
2021	Hu et al. [192]	DCNN	3	300W-LP, AFLW2000, BIWI
2021	Khan et al. [100]	Segmentation based Soft-max classifier	3	AFLW, BU, ICT-3DHP, Pointing'04
2021	Liu et al. [158]	Multi-task DCNN	3	AFLW [◊] , AFLW2000*, WIDER*
2021	Naina Dhingra [193]	DCNN	3	300W-LP, AFLW2000, BIWI
2021	Ruan et al. [164]	Model based 3DMM + DCNN	3	300W-LP ^{◊◊*} , AFLW2000 ^{◊◊*} , Florence [◊]
2021	Sheka et al. [175]	DCNN	3	300W-LP, AFLW, AFLW2000, BIWI
2021	Viet et al. [123]	Multi-task DCNN	3	300W-LP ^{†*} , BIWI ^{†*} , CMU Panoptic ^{†*}
2021	Viet et al. [2]	DCNN	3	300W-LP, AFLW2000, CMU Panoptic, UET-Headpose

2021	Xia et al. [120]	Multi-task DCNN	3	300W-LP*, 300VW \diamond , WFLW \diamond , WIDER \dagger
2021	Xin et al. [190]	Model based Graph Convolutional Network	3	300W-LP, AFLW2000, BIWI
2021	Wu et al. [160]	Model based 3DMM + DCNN	3	300W-LP $\diamond\diamond$ *, 300VW \diamond , AFLW \diamond , AFLW2000 \diamond *, Florence \diamond
2022	Cantarini et al. [186]	Model based DCNN	3	300W-LP, AFLW2000, BIWI
2022	Naina Dhingra [180]	DCNN	3	300W-LP, AFLW2000, BIWI

Table 4.2: Head pose estimation publications most cited in recent literature. For multi-task models we annotated the specific tasks for which each dataset is used as follows: * head pose estimation, \dagger face detection, \diamond face alignment, \star gender classification, \triangle face recognition, \square age estimation, \circ face reconstruction.

4.6 Evaluation pipelines

Currently, in the state-of-the-art works [60–62, 79, 148, 160, 164, 174, 175, 190], there are two primary datasets for training: 300W-LP [4] and BIWI [5], corresponding two main datasets for testing AFLW2000-3D [4] and a part of BIWI [5].

The two most used evaluation protocols are [168]:

- *P1*: Training performed on a single dataset (300W-LP [4]), while BIWI [5] and AFLW2000-3D [4] are used as test sets. Only images with head rotation angles in the range $[-99^\circ, +99^\circ]$ are typically considered (in the case of AFLW2000 31 images are discarded);
- *P2*: Training and test sets are derived from the BIWI dataset [5], in some cases random split is applied (typically, 80% and 20% images), in others split by subject (18 and 2 subjects), recently the most common is the split by sequence (16-8 sequences for training and test respectively), but also n -fold cross-validation and leave-one-out cross-validation are used in the literature.

However, a major drawback of the considered evaluation pipelines is that the head pose angles (including pitch, yaw and roll) are all in the range $[-99^\circ, +99^\circ]$, therefore the prediction of the models are restricted to a “narrow range”, making the models themselves less effective with large-angle data, such as from security cameras [2].

Name	Type	Eval pipeline $P1$ test on BIWI				Eval pipeline $P1$ test on AFLW2000				Eval pipeline $P2$					MB	Param 10 ⁶	Extra training data	Data	Full range	Pre-process. step
		Pitch	Yaw	Roll	MAE	Pitch	Yaw	Roll	MAE	Pitch	Yaw	Roll	MAE	Split						
3DDFA [4]	MB	12.3	36.2	8.78	19.1	8.53	5.40	8.25	7.39									RGB	N	
KEPLER [111]	MT	17.2	8.8	16.2	13.9													RGB	N	
Dlib (68 landmarks) [152]	MB	13.8	16.8	6.19	12.2	13.6	23.1	10.5	15.8						6-24			RGB	N	
FAN (12 points) [153]	MB	7.48	8.53	7.63	7.89	12.3	6.36	8.71	9.12						183	~36.6		RGB	N	
Liu et al. [16]	D									6.10	6.00	5.70	5.90	Rnd				RGB	N	
Drouard et al. [183]	ME									5.90	4.70	4.10	5.20	Sbj				RGB	N	V_j^{fd}
MT-Net v2 (Euler) [123]	MT	7.23	4.64	6.23	6.03					5.33	6.02	5.11	5.48	Seq				RGB	Y	Direct
Shao et al. [197]	D	7.25	4.59	6.15	6.00	6.37	5.07	4.99	5.48						93	24.6		RGB	N	JCFDA ^{fd}
HHP-Net [186]	MB	7.00	4.14	4.40	5.18	10.12	5.26	7.73	7.70	4.79	3.04	3.21	3.68	Seq	0.4	0.1		RGB	N	OpenPose ^{kd}
VGG-IR-FT [187]	D									4.68	3.12	3.07	3.62	Seq				RGB	N	
Hopenet ($\alpha = 2$) [62]	D	6.98	5.17	3.39	5.18	6.56	6.47	5.44	6.16						95.9	23.9		RGB	N	FR ^{fd}
Hopenet ($\alpha = 1$) [62]	D	6.61	4.81	3.27	4.90	6.64	6.92	5.67	6.41	3.39	3.29	3.00	3.23	Seq	95.9	23.9		RGB	N	FR ^{fd}
RetinaFace R50 (5pnt) [60]	D	6.42	4.07	2.97	4.49	9.64	5.10	3.92	6.22									RGB	N	Direct
SSR-Net-MD [168]	D	6.31	4.49	3.61	4.65	7.09	5.14	5.89	6.01	4.35	4.24	4.19	4.26	Seq	1.1	0.2		RGB	N	MTCNN ^{fd}
MT-Net v2 (Vecotr) [123]	MT	4.29	4.62	4.52	4.48					3.90	5.33	3.28	4.17	Seq				RGB	Y	Direct
LwPosr [180]	D	4.87	4.11	3.19	4.05	6.38	4.44	4.88	5.35	4.65	3.62	3.78	4.01	Seq		0.15		RGB	N	MTCNN ^{fd}
FSA-Caps (1 × 1) [168]	D	5.15	4.56	2.94	4.31	6.19	4.82	4.76	5.25						1.1			RGB	N	MTCNN ^{fd}
FSA-Caps-Fusion [168]	D	4.96	4.27	2.76	4.00	6.08	4.50	4.64	5.07	4.29	2.89	3.6	3.6	Seq	5.1	1.2		RGB	N	MTCNN ^{fd}
FDN [188]	D	4.96	4.27	2.76	4.00	6.08	4.50	4.64	5.07	3.98	3.00	2.88	3.29	Seq	5.8			RGB	N	MTCNN ^{fd}
QuatNet [173]	D	5.49	4.01	2.93	4.14	5.61	3.97	3.92	4.50									RGB	N	
HeadPosr EH38 [193]	D	5.10	4.08	3.02	4.06	4.86	4.60	2.87	4.11									RGB	N	MTCNN ^{fd}
HeadPosr EH64 [193]	D	5.44	3.37	2.69	3.83	5.84	4.64	4.30	4.92	4.03	2.59	3.53	3.38	Seq				RGB	N	MTCNN ^{fd}
HR-AT-nBG [192]	D									3.74	3.07	3.11	3.31	Seq				RGB	N	Direct
CNN+Heatmap [147]	MB									3.49	3.46	2.74	3.23	8FCV		3.2		RGB	N	
Gu et al. [11]	RNN									4.03	3.91	3.03	3.66	Seq	500	~136		RGB	N	
Gu et al. [11]	RNN									3.48	3.14	2.60	3.10	Seq	500	~136		RGB+	N	
Hybrid Coarse-fine [189]	D					6.23	4.82	5.14	5.40	2.64	3.43	2.98	3.02	8FCV	96.7	~24		Time RGB	N	FR ^{fd}
img2pose [60]	MT	3.55	4.57	3.24	3.79	5.03	3.43	3.28	3.91								WIDER*	RGB	N	Direct
MNN [119]	MT	4.61	3.98	2.39	3.66 ²	4.69	3.34	3.88	4.42									RGB	N	Direct
Ahn et al. [169]	D									3.40	2.80	2.60	2.93	Rnd				RGB	N	
TriNet [79]	D	4.76	3.05	4.11	3.97	5.77	4.20	4.04	4.67	3.04	2.44	2.93	2.80	3FCV		~26		RGB	N	MTCNN ^{fd}
3DDFA-TPAMI [161]	MB					5.98	4.33	4.30	4.87									RGB	N	FTF ^{fd}
MOS [122]	MT					5.42	3.91	3.98	4.43									RGB	N	Direct
FSA-Net-Wide [2]	D					5.69	4.59	2.85	4.37						2.91		UET, CMU	RGB	Y	
3DDFA-V2 [163]	MB					4.09	3.42	3.48	4.27								UMD	RGB	N	
2DASL [162]	MB					5.06	3.85	3.50	4.13									RGB	N	
SADRNet [164]	MB					5.00	2.93	3.54	3.82						60			RGB	N	
GLDL [196]	D	5.61	4.12	3.14	4.29	5.06	3.03	3.68	3.93									RGB	N	FR ^{fd}
KD-ResNet152 [175]	D	4.73	3.50	2.87	3.70 ³	4.52	2.97	3.48	3.48 ³	2.88	2.61	2.37	2.62	Seq				RGB	N	Yolo-v5 ^{fd}
KD-ResNet18 [175]	D	5.07	3.96	3.06	4.03	4.69	3.00	3.22	3.64	2.82	2.59	2.15	2.58	Seq				RGB	N	Yolo-v5 ^{fd}
Direct Regression [191]	D									2.75	2.64	2.24	2.54	Seq				RGB	N	FR ^{fd}
RankPose [174]	D	4.77	3.59	2.76	3.71	4.75	2.99	3.25	3.66									RGB	N	MTCNN ^{fd}
EVA-GCN [190]	MB	4.78	4.01	2.98	3.92	5.34	4.46	4.11	4.64	2.82	2.01	1.89	2.24	Seq	1.03	~3.3		RGB	N	FAN ^{kd}
WHENet [61]	D	4.39	3.99	3.06	3.81	6.24	5.11	4.92	5.42						17.1	4.4	CMU	RGB	Y	Yolo-v3 ^{fd}
WHENet-V [61]	D	4.10	3.60	2.73	3.48 ¹	5.75	4.44	4.31	4.83						17.1	4.4		RGB	N	Yolo-v3 ^{fd}
SynergyNet [160]	MB					4.09	3.42	2.55	3.35 ²						3.8			RGB	N	Direct
Xia et al. [148]	MB					2.05	0.63	1.70	1.46 ¹	2.52	2.83	2.86	3.74	5FCV				RGB	N	FAN ^{kd}
Fanelli et al. [5]	ML									8.50	7.90	8.90	8.43	Sbj				Depth	N	V_j^{fd}
Baltrusaitis et al. [21]	MB									5.10	11.30	6.30	7.60	Sbj				RGB+D	N	V_j^{fd}
Saeed et al. [182]	ML									5.00	4.30	3.90	4.40	Sbj				RGB+D	N	V_j^{fd}
LMK [134]	ME									3.80	3.60	5.20	4.20	LIO				Depth	N	
DESC [134]	ME									3.40	3.30	3.30	3.33	LIO				Depth	N	
Papazov et al. [181]	MB									2.50	3.80	3.00	3.20					Depth	N	V_j^{fd}
Martin et al. [165]	MB									2.50	2.60	3.60	2.90	Sbj				Depth	Y [∇]	Videmo ^{fd}
Meyer et al. [166]	MB									2.40	2.10	2.10	2.20	Sbj				Depth	N	Custom ^{fd}
Yu et al. [185]	MB									1.53	2.49	2.18	2.07 ³					RGB+D	Y	Dlib ^{fd,kd}
HeadFusion [10]	MB									1.45	2.54	2.10	2.03 ²					RGB+D	Y	Dlib ^{fd,kd}
POSEidon [7]	D									1.60	1.70	1.80	1.70 ¹	Sbj	3.4			Depth	Y	CustomNN ^{fd}

Table 4.3: Evaluation results of head pose estimation on AFLW2000 [4] and BIWI [5]. For the evaluation protocol P2 many variants are reported in the literature: Random split, Split by subject (18 and 2 subjects), Split by sequence (16 and 8 sequences), n -fold cross-validation and Leave-one-out cross-validation, the splitting method is reported here when available. Model type: (D) Deep learning regressor; (MB) Model based; (ME) Manifold embedding; (ML) Machine learning regressor; (MT) Multi-task; (RNN) Recurrent neural network. Narrow range models are optimized for $\pm 99^\circ$, full range for $\pm 180^\circ$, ∇ means $\pm 120^\circ$. Extra training data used are CMU Panoptic [8], UET-Headpose [2], UMDFace [6] and WIDER [1] (* head pose are annotated with a deep learning regressor). In pre-processing *fd* means face detector, *kd* means keypoints (landmarks) detector. VJ is Viola-Jones face detector implemented in openCV [201]; FR is Faster-RCNN [202]; JCFDA [203]; openPose [204]; Yolo [205]; Dlib [152]; FTF is finding tiny faces detector [206]. Other training/testing strategies used for BIWI dataset are presented in table 4.5.

For this reason, other researchers used additional head pose datasets, such as Zhou et al. for training the WHENet model [61], where CMU Panoptic dataset [8] was adopted to increase the amount of data, but also because it provides comprehensive yaw angles in the range $[-179^\circ, +179^\circ]$. This is necessary to obtain a model optimized for the full range (360°) of face orientation, improving a lot the result compared to models trained only with 300W-LP [4]. Albiero et al. [60] instead annotated the WIDER face database [1] using a deep learning regressor, and used it during training to increase the robustness of the model. Recently, Viet et al. [2] released the UET-Headpose dataset, also with uniform yaw angle in the range $\pm 179^\circ$, that can be used as new benchmark dataset for full range models.

Moreover, the semi-automatic pipeline used to label 300W-LP [4] and AFLW2000-3D [4] has been criticised for not producing accurate annotations for extreme poses and occluded faces [153]. Valle et al. [119] re-annotated AFLW2000-3D with poses estimated from the correct landmarks, this led to an improvement in model performance.

Other researchers employ synthetic datasets for training and tested on real ones [11, 16, 176–178]. Kuhnke et al. [177] propose novel benchmark datasets that are derived from BIWI [5] and SynHead [11], namely Biwi+, SynBiwi+, SynHead++. They propose these new datasets because SynHead was rendered using the Euler angles provided by BIWI, but with a different sequence of rotation axes. This rotation order, dissimilar to the BIWI one, causes that several SynHead images and BIWI images with the same label show different head rotations. For this

reason, the reannotated SynHead+ contains SynHead images with correct angles. For every image in the BIWI dataset, SynBiwi+ has 10 corresponding images containing the 10 synthetic head models of SynHead. SynHead++ is the union of SynHead+ and SynBiwi+. To further improve the reproducibility manually collected bboxes for BIWI are provided in Biwi+ dataset.

Another dataset often used in the literature both for training and testing is the AFLW [23], however, there isn't a common evaluation protocol used in the many studies published. The most common is:

- *P3*: Train and test set are defined by a random split, 23.386 images are used for training the model (of which typically 2.000 are employed as validation set) and 1.000 images for testing. More details about other evaluation pipelines for AFLW are in table 4.4.

Name	Type	Train	Test	Evaluation pipeline	Pitch	Yaw	Roll	MAE	Data type	Pre-processing
DLDL (KL) [194]	D	AFLW	AFLW	1	5.75	6.60			RGB	
AVM [130]	ME	AFLW	AFLW	2				17.48	RGB	VJ ^{fd}
Dlib [†] [152]	MB	Not required	AFLW	Unknown	13.6	23.1	10.5	15.7	RGB	Direct
TRFH [124]	MT	AFLW	AFLW	Unknown	23.81	5.49	17.26	15.52	RGB	
FAN [†] [153]	MB	Not required	AFLW	Unknown	12.3	6.4	8.7	9.13	RGB	
3DDFA [†] [4]	MB	Not required	AFLW	Unknown	8.2	5.4	8.7	7.43	RGB	
GLDL [196]	D	AFLW	AFLW	Unknown	5.31	6.00	3.75	5.02	RGB	FR ^{fd}
LeNet-5 [171]	D	AFLW	AFLW	5-FCV	7.15	11.04	4.40	7.53	RGB	OpenPose ^{kd}
MLP+Locations (5pnt.) [147]	MB	AFLW	AFLW	5-FCV	6.64	9.56	4.68	6.96	RGB	
CNN+Heatmaps (5pnt.) [147]	MB	AFLW	AFLW	5-FCV	5.58	6.19	3.76	5.18	RGB	
Segm+CNN [100]	SB	AFLW	AFLW	10-FCV	3.2	4.9			RGB	SSD ^{fd}
HPE-MSF-CRFs [97]	SB	AFLW	AFLW	10-FCV	4.89	4.25	3.20	4.11	RGB	SSD ^{fd}
HAG-MSF-CRFs [99]	SB	AFLW	AFLW	10-FCV	4.89	4.25	3.20	4.11	RGB	SSD ^{fd}
QT-PYR [150]	MB	Not required	AFLW	3	7.60	7.60	7.17	7.45	RGB	VJ ^{fd} , Dlib ^{kd}
Hybrid Coarse-fine [189]	D	300W-LP	AFLW	3	5.38	6.18	5.09	5.55	RGB	
4D-AS [151]	MB	Not required	AFLW	3	4.82	3.11	2.25	3.39	RGB	Dlib ^{kd}
KD-ResNet18 [175]	D	AFLW	AFLW	4	6.02	5.45	4.16	5.21	RGB	Yolo-v5 ^{fd}
KD-ResNet152 [175]	D	AFLW	AFLW	4	5.93	5.41	4.07	5.14	RGB	Yolo-v5 ^{fd}
QuatNet [173]	D	300W-LP	AFLW	5	4.32	3.93	2.59	3.61	RGB	
CCR [195]	MT	AFLW	AFLW	6	5.85	5.22	2.51	4.53	RGB	
KEPLER [145]	MB	AFLW	AFLW	P3	5.85	6.45	8.75	6.45	RGB	Direct FR ^{fd} OpenPose ^{kd}
Hyperface [108]	MT	AFLW	AFLW	P3	6.13	7.61	3.92	5.88	RGB	
Hopenet ($\alpha = 1$) [62]	D	AFLW	AFLW	P3	5.89	6.26	3.82	5.32	RGB	
MLP+Locations (5pnt.) [147]	MB	AFLW	AFLW	P3	5.84	6.02	3.56	5.14	RGB	
VGG-16 [172]	D	AFLW	AFLW	P3	5.24	6.45	3.61	5.10	RGB	
AlexNet [172]	D	AFLW	AFLW	P3	5.21	6.40	3.47	5.02	RGB	
MOS [122]	MT	AFLW	AFLW	P3				4.89	RGB	
ResNet-50 [172]	D	AFLW	AFLW	P3	5.02	6.03	3.22	4.75	RGB	
VGG-19 [172]	D	AFLW	AFLW	P3	4.93	5.99	3.15	4.69	RGB	
ResNet-101 [172]	D	AFLW	AFLW	P3	4.98	5.69	3.07	4.59	RGB	
ResNet-152 [172]	D	AFLW	AFLW	P3	4.88	5.92	2.98	4.58	RGB	
CNN+Heatmaps (5pnt.) [147]	MB	AFLW	AFLW	P3	4.43	5.22	2.53	4.06	RGB	OpenPose ^{kd}
MNN [119]	MT	AFLW	AFLW	P3	3.07	4.16	2.43	3.22	RGB	

Table 4.4: Evaluation results of head pose estimation on AFLW [23] (ordered by training pipeline). † results taken from [58]. Evaluation pipeline: (1) Random split - 15.561 images for training, 7.848 for testing; (2) Random split - 14.000 images for training, 7.041 for testing; (3) Test on all AFLW; (4) First 2.000 images for testing other for training; (5) Train on other dataset, test on 1.000 random sample from AFLW; (6) Random split - 20.000 images for training other for testing; (n -FCV) n -fold cross-validation. Model type: (D) Deep learning regressor; (MB) Model based; (ME) Manifold embedding; (ML) Machine learning regressor; (MT) Multi-task; (RNN) Recurrent neural network; (SB) Segmentation based model. In pre-processing *fd* means face detector, *kd* means keypoints (landmarks) detector. VJ is Viola-Jones face detector implemented in openCV [201]; FR is Faster-RCNN [202]; openPose [204]; Yolo [205]; Dlib [152]; SSD is Single shot multibox detector [207]. Not all papers specify the pre-processing applied, some are direct methods that incorporate a detection phase, other use face crop from gt bbox.

4.7 Discussion

Head pose estimation is an active research field of computer vision. It remains a challenging task due to internal and external conditions and complex imaging feature face [57].

New databases are released every year because deep learning models require a huge quantity of data for training, but especially to overcome limitations of previously released datasets, such as limited head rotation angle ranges, non-uniform distribution of angles, data captured in constraint environment, limited quality of ground-truth annotations, etc.

Almost all most recent databases have annotations for all three rotation angles (*pitch*, *yaw* and *roll*), mainly acquired using depth cameras or optical motion capture systems. This is clearly an improvement with respect to first datasets that, typically, were acquired using direct suggestion or camera array methods, leading to a discrete number of poses and annotations limited to one or two DoF.

The number and the variety of databases for HPE are increasing year after year, their complexity is grown from simple images with flat backgrounds, to more complex scenarios with images acquired in-the-wild. However, a major drawback of the latter type is that pose is typically annotated manually or estimated with neural networks trained on other datasets, leading to inaccuracies in the ground-truth annotations (see for example figure 4.22).

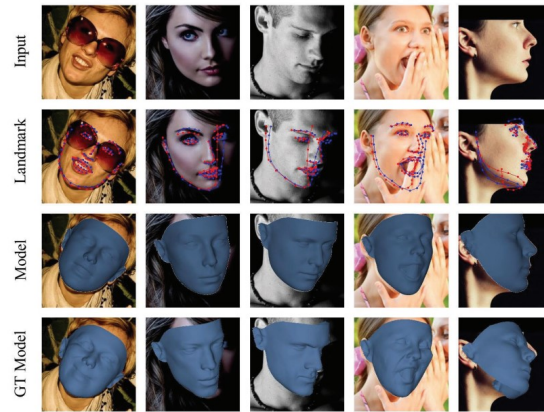


Figure 4.22: Example of inaccuracies in ground-truth annotations on AFLW2000 dataset [4]. In some cases results from SADRNet [164] model are more accurate than the ground-truth. From the top row to the bottom row there are: the AFLW2000 [4] images, the sparse alignment results of SADRNet [164] and the corresponding ground-truth (blue for the former and red the latter), the reconstructed face models of SADRNet [164], and the ground-truth face models [164]. Vall et al. [119] reannotated AFLW2000 with poses estimated from correct landmarks and evaluated their MNN model, the MAE fell from 3.83 to 1.71 after the reannotation (image from [164]).

Among all the databases, Boston University [30] is still used to evaluate head pose estimation methods even if it is one of the oldest, some model-based and segmentation based methods obtain very accurate performance on it, as can be seen in table 4.5. Also Pointing'04 [29] is still employed for research purposes, even if it was introduced back in 2004, due to its challenging nature and high image diversity.

BIWI Kinect [5] has become the de-facto benchmark dataset with a high number of publications that evaluate their models on it. However, this dataset has two main disadvantages: it's a *narrow range* dataset, head rotation angles go from -75° to $+75^\circ$, making it not suitable to evaluate models optimized for *full range* (360°) head rotations; furthermore, it's a dataset with images registered in a constraint environment, therefore less challenging than other that are captured with different

lighting conditions, backgrounds, occlusions and so on.

Nowadays *synthetic databases* [11] [16] [41] enable more precise evaluation and comparison of HPE methods because they contain nearly perfect ground-truth data. However, training solely on synthetic data can cause poor performance when testing on real-world data due to mismatch or shift of underlying data distribution (domain gap). For this reason, training on a combination of synthetic data and real ones can lead to an improvement in the final result, see for example FSA-Net [168] model tested on BIWI dataset [5] in table 4.5.

Name	Type	Train	Test	Evaluation pipeline	Pitch	Yaw	Roll	MAE	MAWE	BMAE	Data type
POSEidon [33]	D	AutoPose	AutoPose	18 sequences for train, 1 for test	2.96	3.16	3.99	3.37		11.86	IR
DLDL (KL) [194]	D	BJUT-3D	BJUT-3D	5-fold cross-validation	0.02	0.07					RGB
FSA-Net Caps-Fusion [178]	D	SynHead	BIWI	Test on all BIWI	8.58	6.04	9.82	8.29			RGB
KPM [158]	MB	Not required	BIWI	Test on all BIWI	7.94	5.81	6.74	6.83			RGB
FSA-Net Caps-Fusion [178]	D	SGD [†] (300k)	BIWI	Test on all BIWI	6.51	5.86	6.63	6.34			RGB
DANN [177]	D	SynHead++	BIWI+	Test on all BIWI+	8.08	6.17	3.91	6.05			RGB
QT-PYR [150]	MB	Not required	BIWI	Test on all BIWI	7.51	4.07	5.50	5.69			RGB
4C-4S [151]	MB	Not required	BIWI	Test on all BIWI	3.95	6.21	4.16	4.77			RGB
DC2F [176]	D	SGD [†] (208k) + BIWI	BIWI	Random split BIWI: 12k train, 3k test	5.48	4.76	4.26	4.54			RGB
FSA-Net Caps-Fusion [178]	D	SGD [†] (300k) + BIWI	BIWI	Random split BIWI: 14k train, 1k test	4.54	4.62	3.33	4.16			RGB
PADACO [177]	D	SynHead++	BIWI+	Test on all BIWI+	4.51	4.11	3.78	4.13			RGB
PADACO [177]	D	SynBiwi+	BIWI+	Test on all BIWI+	4.47	4.11	3.56	4.04			RGB
RT-MT-HPE [115]	MT	BIWI + RCV-Face	BIWI	Random split 37 subjects for training, 10 for test	4.3	3.4	3.6	3.76			RGB
Liu et al. [16]	D	Synthetic	BIWI	Random 30 seq. from synthetic db for training, test on all BIWI	4.3	4.5	2.4	3.73			RGB
DANN [177]	D	SynBiwi+	BIWI+	Test on all BIWI+	3.56	3.43	3.03	3.34			RGB
CCR [184]	MB	Not required	BU	Test on 200 images of 5 subjects with uniform lighting conditions	4.8	5.1	3.3	4.4			RGB
S-FLD-HPE [157]	MB	Not required	BU	Test on 200 images of 5 subjects with uniform lighting conditions	5.3	4.9	3.1	4.4			RGB
CHM+PnP [144]	MB	Not required	BU	Test on 200 images of 5 subjects with uniform lighting conditions	4.58	4.87	2.80	4.08			RGB
EHM+PnP [144]	MB	Not required	BU	Test on 200 images of 5 subjects with uniform lighting conditions	3.39	3.99	2.56	3.31			RGB
HPE-FF [159]	MB	Not required	BU	Test on 200 images of 5 subjects with uniform lighting conditions	3.41	3.90	2.32	3.21			RGB
CLM-Z [21]	MB	BU	BU	Unknown	3.00	3.81	2.08	2.97			RGB+D
OpenFace+PnP [32]	MB	Not required	BU	Test on all BU				2.6			RGB
HPE-MSF-CRFs [97]	SB	BU	BU	10-fold cross-validation	2.9	2.1	2.2	2.4			RGB
HAG-MSF-CRFs [99]	SB	BU	BU	10-fold cross-validation	2.9	2.1	2.2	2.4			RGB
Segm+CNN [100]	SB	BU	BU	10-fold cross-validation	2.0	2.4					RGB
MSE-MR [133]	ME	CAS-PEAL-1	CAS-PEAL-1	5-fold cross-validation	2.3	1.0					RGB
MSE-MR [133]	ME	CAS-PEAL-2	CAS-PEAL-2	5-fold cross-validation	30.6	2.9					RGB
MSE-MR [133]	ME	CMU-Pie	CMU-Pie	5-fold cross-validation		1.9					RGB
Cascade Trees [17]	ML	Dali3DHP	Dali3DHP	Leave-one-out cross-val.	7.69	4.73		6.23			RGB+D

OpenFace+PnP [32]	MB	Not required	DD-Pose	Test on all DD-Pose	4	4	5	9		16	RGB
HeHop [35]	ML	DriveAHead	DriveAHead	First 5 subjects for test, other 15 for train						26.3	Depth
OpenFace+PnP [35]	MB	Not required	DriveAHead	Test on first 5 subjects						20.6	IR
HPN [35]	D	DriveAHead	DriveAHead	First 5 subjects for test, other 15 for train						13.4	IR+D
Meyer et al. [166]	MB	Not required	ETH	Test on all ETH	2.3	2.9		2.6			Depth
Liu et al. [129]	ME	FacePix	FacePix	Leave-one-out cross-val.		3.1					RGB
Balasubramanian et al. [127]	ME	FicePix	FacePix	8-fold cross-validation		1.4					RGB
POSEidon [7]	D	ICT-3DHP	ICT-3DHP	Unknown	5.0	7.1	3.5	5.2			Depth
OpenFace+PnP [32]	MB	Not required	ICT-3DHP	Test on all ICT-3DHP				3.2			RGB
CLM-Z [21]	MB	ICT-3DHP	ICT-3DHP	Unknown	3.14	2.90	3.17	3.07			RGB+D
HPE-MSF-CRFs [97]	SB	ICT-3DHP	ICT-3DHP	10-fold cross-validation	3.2	2.6	2.7	3.0			RGB
HPE-MSF-CRFs [99]	SB	ICT-3DHP	ICT-3DHP	10-fold cross-validation	3.2	2.6	2.7	3.0			RGB
Segm+CNN [100]	SB	ICT-3DHP	ICT-3DHP	10-fold cross-validation	2.3	2.9					RGB
AVM [130]	ME	AFLW	McGill	14k random AFLW images as train, 6833 McGill images as test				16.29			RGB
PointNet [34]	MLP	MDM Corpus	MDM Corpus	39 subjects for train, 10 as validation, 10 for test	6.33	5.84	5.77	5.98			Depth
Reg-CNN [198]	D	Multi-Pie	Multi-Pie	3-fold cross-validation		0.02					RGB
POSEidon [7]	D	Pandora	Pandora	Subjects 10, 14, 16, 20 for test, the other for training	5.7	4.9	9.0	6.53			Depth
KPM [158]	MB	Not required	Pandora	Test on all Pandora	4.99	6.33	3.87	5.06			RGB
Pixel-based segmentation [96]	SB	Pointing'04	Pointing'04	People 1-7 for train, people 8-15 for test		3.75					RGB
Super-pixel segmentation [96]	SB	Pointing'04	Pointing'04	People 1-7 for train, people 8-15 for test		5.69					RGB
Khan et al. [98]	SB	Pointing'04	Pointing'04	10-fold cross-validation		2.79					RGB
Hopenet [158]	MB	Pointing'04 (reannotated)	Pointing'04 (reannotated)	Train-test split unknown	19.59	26.61		23.10			RGB
FSA-Net [158]	MB	Pointing'04 (reannotated)	Pointing'04 (reannotated)	Train-test split unknown	18.01	25.90		21.96			RGB
LeNet-5 [171]	D	Pointing'04	Pointing'04	Leave-one-out cross-val.	10.71	7.74		9.23			RGB
MSE-MR [133]	ME	Pointing'04	Pointing'04	5-fold cross-validation	9.6	8.1		8.85			RGB
4C_4S [151]	MB	Not required	Pointing'04	Test on all Pointing'04	6.34	10.63		8.48			RGB
3DDFA [158]	MB	Not required	Pointing'04 (reannotated)	Test on all Pointing'04	7.38	6.18		6.77			RGB
KPM [158]	MB	Not required	Pointing'04 (reannotated)	Test on all Pointing'04	5.27	4.30		4.78			RGB
DLDL (KL) [194]	D	Pointing'04	Pointing'04	5-fold cross-validation	1.69	3.16		2.43			RGB
HPE-MSF-CRFs [97]	SB	Pointing'04	Pointing'04	10-fold cross-validation	1.32	2.68		1.94			RGB
HAG-MSF-CRFs [99]	SB	Pointing'04	Pointing'04	10-fold cross-validation	1.18	2.32		1.75			RGB
Segm+CNN [100]	SB	Pointing'04	Pointing'04	10-fold cross-validation	1.02	2.02		1.52			RGB
Reg-CNN [198]	D	Pointing'04	Pointing'04	5-fold cross-validation	0.76	1.74		1.25			RGB
LMK [134]	ME	SASE	SASE	28 subjects for training, 12 subjects for testing	7.07	6.50	6.06	6.54			Depth
FSA-Net Caps-Fusion [178]	D	SGD [†]	SASE	Test on all SASE	7.76	6.52	5.61	6.63			RGB
DESC [134]	ME	SASE	SASE	28 subjects for training, 12 subjects for testing	6.64	6.21	4.60	5.82			Depth
FSA-Net Caps-Fusion [178]	D	300W-LP	SASE	Test on all SASE	7.27	5.77	3.72	5.59			RGB
FSA-Net Caps-Fusion [178]	D	SGD [†] + SASE	SASE	Random split SASE: 1k for train, other for test	7.13	5.10	3.64	5.29			RGB
Gu et al. [11]	RNN	SynHead	SynHead	8 subjects for training, 2 for testing	1.55	1.78	1.66	1.66			RGB
Liu et al. [16]	D	Synthetic	Synthetic	Random split: 30 seq. for train, 7 for test	3.4	2.7	2.2	2.76			RGB
MSE-MR [133]	ME	Taiwan	Taiwan	5-fold cross-validation		5.8					RGB
OpenFace+PnP [10]	MB	Not required	UbiPose	Unknown	4.45	9.49	3.83	6.28			RGB
HeadFusion [10]	MB	UbiPose	UbiPose	Unknown	4.37	4.63	3.83	4.28			RGB+D
WHENet [2]	D	300W-LP + CMU Panoptic	UET-Headpose-val						53.65		RGB

FSA-Net-Wide [2]	D	300W-LP	UET- Headpose-val			52.76	RGB
FSA-Net-Wide [2]	D	300W-LP + CMU Panop- tic	UET- Headpose-val			52.72	RGB
FSA-Net-Wide [2]	D	UET- Headpose- train	UET- Headpose-val			9.30	RGB
FSA-Net-Wide [2]	D	300W-LP + CMU Panop- tic + UET- Headpose- train	UET- HEadpose-val			7.29	RGB

Table 4.5: Evaluation results of head pose estimation on other databases. † Syntehtic Generated Data. Model type: (D) Deep learning regressor; (MB) Model based; (ME) Manifold embedding; (ML) Machine learning regressor; (MT) Multi-task; (RNN) Recurrent neural network; (SB) Segmentation based model.

Recently, the most active sub-field seems to be “driver head pose estimation”, in the last five years five public datasets that address this specific scenario have been released, each with thousands or millions of images. This is mainly due to the increasing interest in driving assistance systems that aim to monitor the driver’s attention, behaviour and intention, and the fact that head pose is a key element to obtain accurate results [33] [32].

Another important trend observed is that the number of head pose publications has increased in the past few years. More and more people are interested in this area, leading the development of many different approaches to solve the same problem. Nowadays, deep learning and methods based on convolutional neural networks are the most pervasive, these are used to estimate head pose from monocular images, from a set of detected facial landmarks, from a combination of both in a multi-task approach, or even are used to perform 3D dense face alignment/reconstruction, from which the head pose information is obtained as a by-product.

Segmentation based methods are the only recently developed methods that mainly rely on classical machine learning models. These showed how a strong correlation is present between face parts segmentation and its corresponding pose, and that a higher face segmentation leads to accurate pose estimation and vice versa [97]. However, seems to have not been thoroughly investigated yet, this might be because a severe drop in performance is often registered when segmentation is applied in unconstrained environments [99].

What emerges most from the literature is the strong correlation between face alignment and head pose estimation. This correlation is exploited in different ways in the literature, among the best performing methods:

- Xia et al. [148] perform face alignment and then create a landmark *heatmap* that is given as input (along with the facial image) to a CNN. They obtain the best result on AFLW2000 dataset [4] because the heatmap generator improves the generalization ability by making the CNN focus on the area around facial landmarks and reducing the interference from background significantly. However, this method does not remarkably improve the performance on datasets taken under controllable conditions, such as BIWI [5].
- Valle et al. [119] combine face alignment and head pose estimation in a multi-task model improving the overall performance, obtaining the best result on AFLW dataset [23].
- Xin et al. [190] construct a landmark-connection graph to model the complex non-linear mapping between graph topologies and head pose angles. Their model has the lowest MAE when trained and tested on BIWI dataset [5] among the models that use only RGB data.
- Wu et al. [160] exploit facial landmarks to guide 3D facial geometry learning. Pose in this case is a by-product that a backbone network learns during 3DMM parameter regression. SynergyNet outperforms all deep learning regressors on AFLW2000 dataset [4].

Among the presented methods 3DMM based are very interesting, they focus on face reconstruction and incorporate occlusion aware mechanisms very useful in complex scenarios. Moreover, because these methods do not use any ground-truth head pose label during training, they do not suffer from the inaccuracy of head pose labels which exist in most publicly available training datasets. Room for improvement might exist by designing specialized loss functions and addressing specifically the head pose estimation task.

From table 4.2 we can see that almost all the models can estimate 3 DoF, actually some of them (such as 3DMM based) can estimate 6 DoF, but databases are mainly recorded for 3 DoF or less. This highlights a great evolution, indeed until a few years ago researchers focused more on yaw estimation, because of its importance in applications such as human attention, gaze estimation, etc. Deep learning changed the trend, all three rotation angles are currently being addressed in most works.

From table 4.3 we can see how methods that use *depth* data alone, or both *RGB* and *depth* information, can usually achieve better results. The use of depth data enhances the efficacy under challenging illumination conditions and occlusions, making the models suitable to particular challenging contexts, such as automotive. From table 4.5 we can see that recently also thermal infrared images (IR) are used as input for HPE algorithms, in some cases obtaining better results than with depth information. However, depth or infrared data are not always available in real-world contexts, therefore methods based only on monocular images have more generalization abilities and usually less computational costs.

We also saw that different representations are exploited in the literature, the majority of methods use the Euler representation [16, 62, 74, 119, 147, 150, 151, 168, 182, 183, 187, 188, 196, 196, 197], others use rotation matrix [60, 79, 134, 143, 148, 166, 181, 185] and in one case quaternions are exploited [173]. Since different datasets may have different annotations for the angles, to test the methods a representation is usually chosen and, by the transformation formulas, the labels of the dataset are normalized accordingly. We do not notice a significant difference in terms of performances using one representation rather than another.

The main problem that emerges from this analysis is that different experimental set-ups and different validation protocols are adopted for HPE algorithms, and this strongly influences the final result, making comparison difficult. Different evaluation pipelines (see section 4.6) and different pre-processing methods are used (see tables 4.3, 4.4). We can categorize pre-processing techniques into three categories: face detection, face alignment and face segmentation. In the face detection case, for longer, a very popular technique has been the Viola-Jones method [201], then

the most used became MTCNN [200], but there is no uniformity in literature. Among the face alignment methods most used are Dlib [152] and FAN [153]. Segmentation was recently introduced in the field and for now it's little used and not thoroughly investigated.

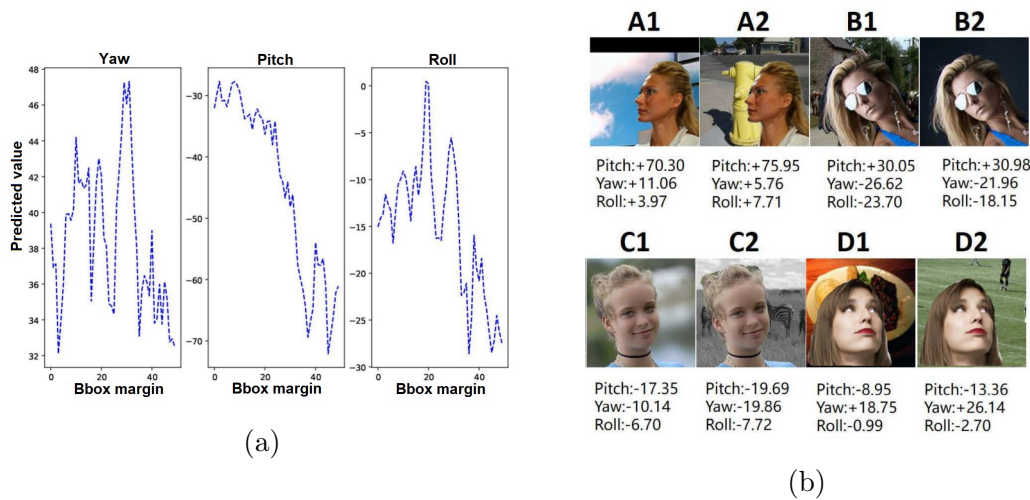


Figure 4.23: Influence of bbox margin and background on head pose estimation: (a) Influence of bbox margin on head pose estimation. The values predicted by FSA-Net [168] change significantly with the change of bounding box size on all three axes. The network is not robust to the change of bbox margin; (b) Influence of background on head pose estimation. The values predicted by SSR-Net-MD [168] are not robust in different background, e.g. the offset of pitch and yaw between A1 and A2 is about 5° (images from [199]).

Shao et al. [197] discovered in their experiments that bounding box margin has a large impact on the final accuracy of the model; head pose estimators are vulnerable to changes in the background scene around the target face, as shown in image 4.23. To solve this problem Xue et al. [199] proposed a convolutional cropping module (CCM) that can learn to crop the input image to an attentional area for head pose regression, and a background augmentation technique that can make the network more robust to the background noise. In their experiment SSR-Net-MD [168] MAE error fell from 6.01 to 5.38 and FSA-Net [168] goes from 5.25

to 5.13 thanks to CCM and background augmentation. If on one hand this shows how there are techniques that allow to improve the results obtained, on the other hand differences in the ways of getting the bounding boxes do not allow for a valid comparison of the methods for HPE.

The same problem emerged for face landmark detectors, as shown by Xin et al. [190] in their experiments (see table 4.6). Model-based methods that use different landmark detectors can get better or worst results mainly due to keypoints detector accuracy.

Multi-task approaches have the advantage that can be trained and optimized to solve multiple tasks, reducing the inference of the pre-processing phase.

Landmark detector	Pitch	Yaw	Roll	MAE
EVA-GCN + OpenPose	5.52	7.25	4.78	5.85
EVA-GCN + Dlib	5.76	6.39	3.63	5.26
EVA-GCN + RetinaFace	5.33	5.02	4.26	4.87
EVA-GCN + FAN	5.34	4.96	4.11	4.64
EVA-GCN + GT*	4.15	3.23	3.05	3.48

Table 4.6: Comparisons of different landmark detectors for EVA-GCN performance. GT* means ground-truth data (table from [190]).

The last question that arises is about the evaluation metrics used, MAE is the standard evaluation metric employed, but is optimal only for narrow range models, as explained in section 4.4. It’s worth noting that also Cao et al. [79] criticised the use of MAE of Euler angles as evaluation metric, as according to them it cannot correctly measure the performance on profile images. They proposed to use the Mean Absolute Error of Vectors (MAEV) to assess the performance. They used three vectors, extracted from the rotation matrix, to describe head poses and computed the difference between the ground-truth vectors and the predicted ones. They showed how this representation is more consistent and how MAEV is a more reliable indicator for the evaluation of pose estimation results (see figure 4.24).

For three reasons for us, instead, MAWE (details in section 4.4) could be a

better choice: first, it can be used with Euler angles representation; second, if used to evaluate narrow range methods gives the same result as MAE; third, at this point narrow range methods have reached very high accuracy and it seems the time has come for a switch to full range methods with MAWE as main evaluation metric.

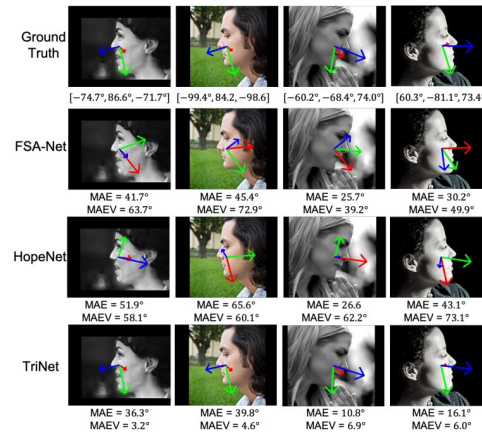


Figure 4.24: Comparison of pose estimation results with MAE and MAEV evaluation metrics on AFLW2000 profile images. All models are trained on 300W-LP (image from [79]).

We also suggest using manually annotated bounding boxes released in Biwi+ [177], when the evaluation pipelines P1 and P2 are used, this can make the models independent from the different results obtained with the various face detectors.

We have seen how deep learning has now become predominant in the field, for the future we expect it to continue like this, however with the development of new architectures that aim to obtain excellent results, but reducing the computational cost.

We expect more and more investigation of some techniques already partially analysed in the literature, such as domain adaption, partial domain adaption, inaccurate semi-supervised learning, knowledge transfer motivated by the fact that obtaining accurate HPE ground-truth is difficult.

For the same reason, we expect an increased use of multi-task learning, which has seen a strong development from 2017 to today. Indeed, head pose can be

used as principal task, but also to enhance the function of some other face-related tasks, including gender classification, expression detection and identity recognition, increasing the amount of available data.

For deformable models, an important improvement will be the ability to selectively ignore parts of the model that are self-occluded, overcoming a fundamental limitation in an otherwise very promising category, especially in unconstrained conditions.

Another interesting direction, not explored yet, is the use of deep learning in segmentation based methods. A possibility is to use convolutional neural networks to regress pose angles from segmented faces, or alternatively segmentation based methods can be extended through geometric/deformable methods, where the feature extraction and classification could exploit specific deep learning architectures.

Although general head pose estimation will continue to be an exciting field with a lot of room for improvement, we expect the development of specific sub-fields that address specific areas of application, such as the “security and surveillance” problem, recently addressed with the release of GOTCHA-I [42] database, or the “driver head pose estimation” which is already an active field [32–35, 48, 146]. Indeed, the role of head pose estimation in driving systems is becoming more and more important. By monitoring the head pose of the driver in real-time and analysing the behaviour of the driver, it will be possible to determine whether the driving status of the driver is good, having a profound impact on the future of automotive safety.

We expect new datasets will continue to be released with an increasing focus on 6 degrees of freedom and full range head angles, thanks to the development of new cheap and powerful RGB-D cameras (such as Microsoft Kinect), and other acquisition techniques. These will guide the future development of the field.

Chapter 5

Conclusion

Head pose estimation is a very important task for human-computer interaction since it provides rich information about the intent, motivation and visual attention of people.

Despite the extensive research in this field, especially during the last years, HPE still remains challenging when images are collected under unconstrained conditions.

In this thesis, we presented a detailed list of publicly available databases, their characteristics and acquisition techniques.

An in-depth survey of head pose estimation methods has been done, by briefly describing oldest and no more used classical approaches, and then providing an extensive analysis of recently proposed approaches, mainly based on deep learning. Indeed, most current heads pose estimation methods exploit convolutional neural networks, from direct regressors to deformable based approaches passing through multi-task learning. We have also presented a comparative analysis of the state-of-the-art performance obtained so far in the field by providing organized and informative tables.

We also listed possible directions for future work. In particular, we expect the introduction of new light DL architectures that can perform well on challenging datasets, i.e., those collected in unconstrained environments. Another interesting direction will be the combination of segmentation based methods with deep learning, but also deformable methods seems to be very interesting.

We also expect the development of new sub-fields with dedicated databases and evaluation pipelines, such as the “driver head pose estimation” that is already very active.

An important trend observed is that the number of head pose publications has increased in the past few years. This is a sign that more and more people are interested in this area, which means that the development cycle of new methods will be faster. A constant and periodic updating of the literature is therefore important.

We hope that this survey thesis help to clarify the evolution of the field, evaluation methodologies and techniques thanks to the provided comprehensive list of datasets, methods and algorithms. This work can be used as starting point for those new to the field that want to orient themselves in it.

Bibliography

- [1] YANG, Shuo, et al. Wider face: A face detection benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 5525-5533.
- [2] VIET, Linh Nguyen, et al. UET-Headpose: A sensor-based top-view head pose dataset. In: 2021 13th International Conference on Knowledge and Systems Engineering (KSE). IEEE, 2021. p. 1-7.
- [3] CAO, Qiong, et al. Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). IEEE, 2018. p. 67-74.
- [4] ZHU, Xiangyu, et al. Face alignment across large poses: A 3d solution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 146-155.
- [5] FANELLI, Gabriele, et al. Real time head pose estimation from consumer depth cameras. In: Joint pattern recognition symposium. Springer, Berlin, Heidelberg, 2011. p. 101-110.
- [6] BANSAL, Ankan, et al. Umdfaces: An annotated face dataset for training deep networks. In: 2017 IEEE international joint conference on biometrics (IJCB). IEEE, 2017. p. 464-473.
- [7] BORGHI, Guido, et al. Poseidon: Face-from-depth for driver pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 4661-4670.

-
- [8] JOO, Hanbyul, et al. Panoptic studio: A massively multiview system for social motion capture. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. p. 3334-3342.
- [9] LI, Peipei, et al. M2FPA: A multi-yaw multi-pitch high-quality dataset and benchmark for facial pose analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. p. 10043-10051.
- [10] YU, Yu; MORA, Kenneth Alberto Funes; ODOBEZ, Jean-Marc. HeadFusion: 360° Head Pose Tracking Combining 3D Morphable Model and 3D Reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 2018, 40.11: 2653-2667.
- [11] GU, Jinwei, et al. Dynamic facial analysis: From bayesian filtering to recurrent neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1548-1557.
- [12] I. LÁzsi et al., "Joint Challenge on Dominant and Complementary Emotion Recognition Using Micro Emotion Features and Head-Pose Estimation: Databases," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 2017, pp. 809-813, doi: 10.1109/FG.2017.102.
- [13] LUSI, Iris; ESCARELA, Sergio; ANBARJAFARI, Gholamreza. Sase: Rgb-depth database for human head pose estimation. In: European conference on computer vision. Springer, Cham, 2016. p. 325-336.
- [14] ARIZ, Mikel, et al. A novel 2D/3D database with automatic face annotation for head tracking and pose estimation. *Computer Vision and Image Understanding*, 2016, 148: 201-210.
- [15] LIU, Yuanyuan, et al. Robust head pose estimation using Dirichlet-tree distribution enhanced random forests. *Neurocomputing*, 2016, 173: 42-53.
- [16] X. Liu, W. Liang, Y. Wang, S. Li and M. Pei, "3D head pose estimation with convolutional neural network trained on synthetic images," 2016 IEEE

- International Conference on Image Processing (ICIP), 2016, pp. 1289-1293, doi: 10.1109/ICIP.2016.7532566.
- [17] S. Tulyakov, R. -L. Vieri, S. Semeniuta and N. Sebe, "Robust Real-Time Extreme Head Pose Estimation," 2014 22nd International Conference on Pattern Recognition, 2014, pp. 2263-2268, doi: 10.1109/ICPR.2014.393.
- [18] DEWANTARA, Bima Sena Bayu; MIURA, Jun. The AISL head orientation database and preliminary evaluations. In: 2015 International Electronics Symposium (IES). IEEE, 2015. p. 140-144.
- [19] GAO, Wen, et al. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2007, 38.1: 149-161.
- [20] DEMIRKUS, Meltem; CLARK, James J.; ARBEL, Tal. Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications*, 2014, 70.1: 495-523.
- [21] T. Baltrusaitis, P. Robinson and L. -P. Morency, "3D Constrained Local Model for rigid and non-rigid facial tracking," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2610-2617, doi: 10.1109/CVPR.2012.6247980.
- [22] ZHU, Xiangxin; RAMANAN, Deva. Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012. p. 2879-2886.
- [23] KOESTINGER, Martin, et al. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: 2011 IEEE international conference on computer vision workshops (ICCV workshops). IEEE, 2011. p. 2144-2151.
- [24] GROSS, Ralph, et al. Multi-pie. *Image and vision computing*, 2010, 28.5: 807-813.

- [25] SAVRAN, Arman, et al. Bosphorus database for 3D face analysis. In: European workshop on biometrics and identity management. Springer, Berlin, Heidelberg, 2008. p. 47-56.
- [26] YIN, Baocai, et al. BJUT-3D large scale 3D face database and information processing. *Journal of Computer Research and Development*, 2009, 46.6: 1009.
- [27] M. D. Breitenstein, D. Kuettel, T. Weise, L. van Gool and H. Pfister, "Real-time face pose estimation from single range images," 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587807.
- [28] G. Little, S. Krishna, J. Black and S. Panchanathan, "A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle," *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, pp. ii/89-ii/92 Vol. 2, doi: 10.1109/ICASSP.2005.1415348.
- [29] GOURIER, Nicolas; HALL, Daniela; CROWLEY, James L. Estimating face orientation from robust detection of salient facial features. In: *ICPR International Workshop on Visual Observation of Deictic Gestures*. 2004.
- [30] M. La Cascia, S. Sclaroff and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322-336, April 2000, doi: 10.1109/34.845375.
- [31] Ba, Sileye; Odobez, Jean-Marc; *IDIAP: A Video Database for Head Pose Tracking Evaluation*; 2005.
- [32] ROTH, Markus; GAVRILA, Darius M. Dd-pose-a large-scale driver head pose benchmark. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019. p. 927-934.

-
- [33] SELIM, Mohamed, et al. AutoPOSE: Large-scale Automotive Driver Head Pose and Gaze Dataset with Deep Head Orientation Baseline. In: VISIGRAPP (4: VISAPP). 2020. p. 599-606.
- [34] JHA, Sumit, et al. The multimodal driver monitoring database: A naturalistic corpus to study driver attention. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [35] SCHWARZ, Anke, et al. Driveahead-a large-scale driver head pose dataset. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017. p. 1-10.
- [36] WALTER, Steffen, et al. The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In: *2013 IEEE international conference on cybernetics (CYBCO)*. IEEE, 2013. p. 128-131.
- [37] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397-403, doi: 10.1109/ICCVW.2013.59.
- [38] SMITH, Brian A., et al. Gaze locking: passive eye contact detection for human-object interaction. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 2013. p. 271-280.
- [39] SIM, Terence; BAKER, Simon; BSAT, Maan. The CMU pose, illumination, and expression (PIE) database. In: *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE, 2002. p. 53-58.
- [40] CHEN, Ju-Chin; LIEN, Jenn-Jier James. A view-based statistical system for multi-view face detection and pose estimation. *Image and Vision Computing*, 2009, 27.9: 1252-1271.

- [41] WERNER, Philipp; SAXEN, Frerk; AL-HAMADI, Ayoub. Landmark based head pose estimation benchmark and method. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 3909-3913.
- [42] BARRA, Paola, et al. Gotcha-I: A multiview human videos Dataset. In: International Symposium on Security in Computing and Communication. Springer, Singapore, 2019. p. 213-224.
- [43] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992, doi: 10.1109/34.121791.
- [44] TU, Jilin, et al. Evaluation of head pose estimation for studio data. In: International Evaluation Workshop on Classification of Events, Activities and Relationships. Springer, Berlin, Heidelberg, 2006. p. 281-290.
- [45] NG, Jeffrey; GONG, Shaogang. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In: Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No. PR00378). IEEE, 1999. p. 14-21.
- [46] L. -. Morency, A. Rahimi and T. Darrell, "Adaptive view-based appearance models," 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., 2003, pp. I-I, doi: 10.1109/CVPR.2003.1211435.
- [47] MURPHY-CHUTORIAN, Erik; DOSHI, Anup; TRIVEDI, Mohan Manubhai. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In: 2007 IEEE intelligent transportation systems conference. IEEE, 2007. p. 709-714.
- [48] M. I. Perdana, W. Anggraeni, H. A. Sidharta, E. M. Yuniarno and M. H. Purnomo, "Early Warning Pedestrian Crossing Intention From Its Head

- Gesture using Head Pose Estimation,” 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2021, pp. 402-407, doi: 10.1109/ISITIA52817.2021.9502231.
- [49] N. Zhuang, Y. Yan, S. Chen and H. Wang, ”Multi-task Learning of Cascaded CNN for Facial Attribute Classification,” 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 2069-2074, doi: 10.1109/ICPR.2018.8545271.
- [50] ZHENG, Xin, et al. A survey of deep facial attribute analysis. *International Journal of Computer Vision*, 2020, 128.8: 2002-2034.
- [51] SENGUPTA, Soumyadip, et al. Frontal to profile face verification in the wild. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016. p. 1-9.
- [52] WANG, Xiang; WANG, Kai; LIAN, Shiguo. A survey on face data augmentation for the training of deep neural networks. *Neural computing and applications*, 2020, 1-29.
- [53] EGGER, Bernhard, et al. 3d morphable face models - past, present, and future. *ACM Transactions on Graphics (TOG)*, 2020, 39.5: 1-38.
- [54] NETO, Euclides N. Arcoverde, et al. Real-time head pose estimation for mobile devices. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, Berlin, Heidelberg, 2012. p. 467-474.
- [55] MURPHY-CHUTORIAN, Erik; TRIVEDI, Mohan Manubhai. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2008, 31.4: 607-626.
- [56] CZUPRYNSKI, Blazej; STRUPCZEWSKI, Adam. High accuracy head pose tracking survey. In: *International Conference on Active Media Technology*. Springer, Cham, 2014. p. 407-420.

-
- [57] SHAO, Xiaofeng, et al. A survey of head pose estimation methods. In: 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). IEEE, 2020. p. 787-796.
- [58] KHAN, Khalil, et al. Head pose estimation: A survey of the last ten years. *Signal Processing: Image Communication*, 2021, 99: 116479.
- [59] JAIN, Anil K.; PARK, Unsang. Facial marks: Soft biometric for face recognition. In: 2009 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009. p. 37-40.
- [60] ALBIERO, Vitor, et al. img2pose: Face alignment and detection via 6dof, face pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. p. 7617-7627.
- [61] ZHOU, Yijun; GREGSON, James. WHENet: Real-time Fine-Grained Estimation for Wide Range Head Pose. arXiv preprint arXiv:2005.10353, 2020.
- [62] RUIZ, Nataniel; CHONG, Eunji; REHG, James M. Fine-grained head pose estimation without keypoints. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018. p. 2074-2083.
- [63] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep learning. MIT press, 2016.
- [64] COOTES, Timothy F., et al. Active shape models-their training and application. *Computer vision and image understanding*, 1995, 61.1: 38-59.
- [65] COOTES, Timothy F. ; EDWARDS, Gareth J. ; TAYLOR, Christopher J. . Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 2001, 23.6: 681-685.
- [66] Euler angles. Wikipedia. https://en.wikipedia.org/wiki/Euler_angles. View January 2022.

-
- [67] ZAHAROVSKIKH, Anastasiya. Head Pose Estimation with Computer Vision. InDataLabs. <https://indatalabs.com/blog/head-pose-estimation-with-cv>. 19 January 2021.
- [68] DENG, Jia, et al. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009. p. 248-255.
- [69] HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.
- [70] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [71] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, 25.
- [72] TAN, Mingxing; LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR, 2019. p. 6105-6114.
- [73] WEIDENBACHER, Ulrich, et al. A comprehensive head pose and gaze database. In: 2007 3rd IET International Conference on Intelligent Environments. IET, 2007. p. 455-458.
- [74] DROUARD, Vincent, et al. Robust head-pose estimation based on partially-latent mixture of linear regressions. IEEE Transactions on Image Processing, 2017, 26.3: 1428-1440.
- [75] DEMENTHON, Daniel F.; DAVIS, Larry S. Model-based object pose in 25 lines of code. International journal of computer vision, 1995, 15.1-2: 123-141.

- [76] HESCH, Joel A.; ROUMELIOTIS, Stergios I. A direct least-squares (DLS) method for PnP. In: 2011 International Conference on Computer Vision. IEEE, 2011. p. 383-390.
- [77] X. Hui, "A Survey for 2D and 3D Face Alignment," 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2019, pp. 57-63, doi: 10.1109/MLBDBI48998.2019.00019.
- [78] KOSTAYAEV, Dimitry. Better rotation representations for accurate pose estimation. Towards data science. 2020. <https://towardsdatascience.com/better-rotation-representations-for-accurate-pose-estimation-e890a7e1317f>
- [79] CAO, Zhiwen, et al. A vector-based representation to enhance head pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021. p. 1188-1197.
- [80] ITOH, Takeshi D., et al. Towards generation of visual attention map for source code. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2019. p. 951-954.
- [81] S. Wu, J. Liang and J. Ho, "Head pose estimation and its application in TV viewers' behavior analysis," 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2016, pp. 1-6, doi: 10.1109/CCECE.2016.7726649.
- [82] K. Smith, S. O. Ba, J. -M. Odobez and D. Gatica-Perez, "Tracking the Visual Focus of Attention for a Varying Number of Wandering People," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1212-1229, July 2008, doi: 10.1109/TPAMI.2007.70773.
- [83] Y. Yamaura, Y. Tsuboshita and T. Onishi, "Head Pose Estimation for an Omnidirectional Camera Using a Convolutional Neural Network," 2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2018, pp. 1-5, doi: 10.1109/IVMSPW.2018.8448756.

-
- [84] WANG, Xiaogang. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 2013, 34.1: 3-19.
- [85] BENFOLD, Ben; REID, Ian. Guiding visual surveillance by tracking human attention. In: *BMVC*. 2009. p. 7.
- [86] K. Sankaranarayanan, M. Chang and N. Krahnstoever, "Tracking gaze direction from far-field surveillance cameras," 2011 IEEE Workshop on Applications of Computer Vision (WACV), 2011, pp. 519-526, doi: 10.1109/WACV.2011.5711548.
- [87] BA, Siley O.; ODOBEZ, Jean-Marc. A study on visual focus of attention recognition from head pose in a meeting room. In: *International Workshop on Machine Learning for Multimodal Interaction*. Springer, Berlin, Heidelberg, 2006. p. 75-87.
- [88] THOMAS, Chinchu; JAYAGOPI, Dinesh Babu. Predicting student engagement in classrooms using facial behavioral cues. In: *Proceedings of the 1st ACM SIGCHI international workshop on multimodal interaction for education*. 2017. p. 33-40.
- [89] AFROZE, Sadia; HOQUE, Mohammed Moshikul. Classification of attentional focus based on head pose in multi-object scenario. In: *International Conference on Intelligent Computing & Optimization*. Springer, Cham, 2019. p. 349-360.
- [90] LANGTON, Stephen RH; BRUCE, Vicki. You must see the point: automatic processing of cues to the direction of social attention. *Journal of Experimental Psychology: Human Perception and Performance*, 2000, 26.2: 747.
- [91] VALENTI, Roberto; SEBE, Nicu; GEVERS, Theo. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 2011, 21.2: 802-815.

-
- [92] MUNHALL, Kevin G., et al. Visual prosody and speech intelligibility: Head movement improves auditory speech perception. *Psychological science*, 2004, 15.2: 133-137.
- [93] MORENCY, Louis-Philippe, et al. Head gestures for perceptual interfaces: The role of context in improving recognition. *Artificial Intelligence*, 2007, 171.8-9: 568-585.
- [94] WANG, Kang; ZHAO, Rui; JI, Qiang. Human computer interaction with head pose, eye gaze and body gestures. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, 2018. p. 789-789.
- [95] A. Grinshpoon, S. Sadri, G. J. Loeb, C. Elvezio and S. K. Feiner, "Hands-Free Interaction for Augmented Reality in Vascular Interventions," 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2018, pp. 751-752, doi: 10.1109/VR.2018.8446259.
- [96] K. Khan, M. Mauro, P. Migliorati and R. Leonardi, "Head pose estimation through multi-class face segmentation," 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 175-180, doi: 10.1109/ICME.2017.8019521.
- [97] KHAN, Khalil, et al. A framework for head pose estimation and face segmentation through conditional random fields. *Signal, Image and Video Processing*, 2020, 14.1: 159-166.
- [98] BENINI, Sergio, et al. Face analysis through semantic face segmentation. *Signal Processing: Image Communication*, 2019, 74: 21-31.
- [99] Khan, K.; Attique, M.; Syed, I.; Sarwar, G.; Irfan, M.A.; Khan, R.U. A Unified Framework for Head Pose, Age and Gender Classification through End-to-End Face Segmentation. *Entropy* 2019, 21, 647.

- [100] KHAN, Khalil, et al. 3D Head Pose Estimation through Facial Features and Deep Convolutional Neural Networks. *Comput. Mater. Contin.*, 2021, 66: 1757-1770.
- [101] HUANG, Gary B.; NARAYANA, Manjunath; LEARNED-MILLER, Erik. Towards unconstrained face recognition. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2008. p. 1-8.
- [102] NG, Jeffrey; GONG, Shaogang. Composite support vector machines for detection of faces across views and pose estimation. *Image and Vision Computing*, 2002, 20.5-6: 359-368.
- [103] HUANG, Jeffrey; SHAO, Xuhui; WECHSLER, Harry. Face pose discrimination using support vector machines (SVM). In: Proceedings. fourteenth international conference on pattern recognition (Cat. No. 98EX170). IEEE, 1998. p. 154-156.
- [104] ZHANG, Zhenqiu, et al. Head pose estimation in seminar room using multi view face detectors. In: International evaluation workshop on classification of events, activities and relationships. Springer, Berlin, Heidelberg, 2006. p. 299-304.
- [105] YAN, Yan, et al. No matter where you are: Flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In: Proceedings of the IEEE international conference on computer vision. 2013. p. 1177-1184.
- [106] YAN, Yan, et al. Evaluating multi-task learning for multi-view head-pose classification in interactive environments. In: 2014 22nd international conference on pattern recognition. IEEE, 2014. p. 4182-4187.
- [107] ZHANG, Zhanpeng, et al. Facial landmark detection by deep multi-task learning. In: European conference on computer vision. Springer, Cham, 2014. p. 94-108.

- [108] RANJAN, Rajeev; PATEL, Vishal M.; CHELLAPPA, Rama. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2017, 41.1: 121-135.
- [109] R. Ranjan, S. Sankaranarayanan, C. D. Castillo and R. Chellappa, "An All-In-One Convolutional Neural Network for Face Analysis," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 2017, pp. 17-24, doi: 10.1109/FG.2017.137.
- [110] XU, Xiang; KAKADIARIS, Ioannis A. Joint head pose estimation and face alignment framework using global and local CNN features. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017. p. 642-649.
- [111] KUMAR, Amit; ALAVI, Azadeh; CHELLAPPA, Rama. Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In: 2017 12th IEEE international conference on automatic face & gesture recognition (fg 2017). IEEE, 2017. p. 258-265.
- [112] CHEN, Jun-Cheng, et al. A real-time multi-task single shot face detector. In: 2018 25th IEEE international conference on image processing (ICIP). IEEE, 2018. p. 176-180.
- [113] CAI, Zhenni, et al. Joint head pose estimation with multi-task cascaded convolutional networks for face alignment. In: 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018. p. 495-500.
- [114] HONG, Chaoqun, et al. Multimodal face-pose estimation with multitask manifold deep learning. *IEEE Transactions on Industrial Informatics*, 2018, 15.7: 3952-3961.
- [115] AHN, Byungtae, et al. Real-time head pose estimation using multi-task deep neural network. *Robotics and Autonomous Systems*, 2018, 103: 1-12.

- [116] WU, Hao; ZHANG, Ke; TIAN, Guohui. Simultaneous face detection and pose estimation using convolutional neural network cascade. *IEEE Access*, 2018, 6: 49563-49575.
- [117] ZHANG, Chen, et al. A privacy-preserving multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *Frontiers in neurorobotics*, 2020, 13: 112.
- [118] EWAISHA, Mahmoud, et al. End-to-End Multitask Learning for Driver Gaze and Head Pose Estimation. *Electronic Imaging*, 2020, 2020.16: 110-1-110-6.
- [119] VALLE, Roberto; BUENAPOSADA, Jose Miguel; BAUMELA, Luis. Multi-task head pose estimation in-the-wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [120] XIA, Jiahao, et al. An Efficient Multitask Neural Network for Face Alignment, Head Pose Estimation and Face Tracking. *arXiv preprint arXiv:2103.07615*, 2021.
- [121] FARD, Ali Pourramezan; ABDOLLAHI, Hojjat; MAHOOR, Mohammad. ASMNet: A Lightweight Deep Neural Network for Face Alignment and Pose Estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. p. 1521-1530.
- [122] LIU, Yepeng, et al. MOS: A Low Latency and Lightweight Framework for Face Detection, Landmark Localization, and Head Pose Estimation. *arXiv preprint arXiv:2110.10953*, 2021.
- [123] VIET, Hoang Nguyen, et al. Simultaneous face detection and 360 degree head pose estimation. In: *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2021. p. 1-7.
- [124] CHEN, Shicun, et al. TRFH: towards real-time face detection and head pose estimation. *Pattern Analysis and Applications*, 2021, 24.4: 1745-1755.

- [125] MCKENNA, Stephen J.; GONG, Shaogang. Real-time face pose estimation. *Real-Time Imaging*, 1998, 4.5: 333-347.
- [126] RAYTCHEV, Bisser; YODA, Ikushi; SAKAUE, Katsuhiko. Head pose estimation by nonlinear manifold learning. In: *Proceedings of the 17th International Conference on Pattern Recognition*, 2004. ICPR 2004. IEEE, 2004. p. 462-466.
- [127] BALASUBRAMANIAN, Vineeth Nallure; YE, Jieping; PANCHANATHAN, Sethuraman. Biased manifold embedding: A framework for person-independent head pose estimation. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007. p. 1-7.
- [128] HUANG, Dong, et al. Supervised local subspace learning for continuous head pose estimation. In: *CVPR 2011*. IEEE, 2011. p. 2921-2928.
- [129] LIU, Xiangyang; LU, Hongtao; LI, Wenbin. Multi-manifold modeling for head pose estimation. In: *2010 IEEE international conference on image processing*. IEEE, 2010. p. 3277-3280.
- [130] SUNDARARAJAN, Kalaivani; WOODARD, Damon L. Head pose estimation in the wild using approximate view manifolds. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2015. p. 50-58.
- [131] WANG, Chao; GUO, Yuanhao; SONG, Xubo. Head pose estimation via manifold learning. *Manifolds-Current Research Areas*, 2017.
- [132] PENG, Xi, et al. Head pose estimation by instance parameterization. In: *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014. p. 1800-1805.
- [133] DIAZ-CHITO, Katerine, et al. Continuous head pose estimation using manifold subspace embedding and multivariate regression. *IEEE Access*, 2018, 6: 18325-18334.

- [134] DERKACH, Dmytro; RUIZ, Adria; SUKNO, Federico M. Tensor decomposition and non-linear manifold modeling for 3D head pose estimation. *International Journal of Computer Vision*, 2019, 127.10: 1565-1585.
- [135] YAO, Pingping; EVANS, Glyn; CALWAY, Andrew. Using affine correspondence to estimate 3-d facial pose. In: *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*. IEEE, 2001. p. 919-922.
- [136] OHAYON, Shay; RIVLIN, Ehud. Robust 3d head tracking using camera pose estimation. In: *18th International Conference on Pattern Recognition (ICPR'06)*. IEEE, 2006. p. 1063-1066.
- [137] LU, Le, et al. Model and exemplar-based robust head pose tracking under occlusion and varying expression. In: *Proc. of CVPR*. 2001.
- [138] MALCIU, Marius; PRETEUX, Françoise. A robust model-based approach for 3d head tracking in video sequences. In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE, 2000. p. 169-174.
- [139] HORPRASERT, Thanarat; YACOOB, Yaser; DAVIS, Larry S. Computing 3-d head orientation from a monocular image sequence. In: *Proceedings of the second international conference on automatic face and gesture recognition*. IEEE, 1996. p. 242-247.
- [140] WANG, Jian-Gang; SUNG, Eric. EM enhancement of 3D head pose estimated by point at infinity. *Image and Vision Computing*, 2007, 25.12: 1864-1874.
- [141] SAPIENZA, Michael; CAMILLERI, K. *Fasthpe: A recipe for quick head pose estimation*. Systems and Control Engineering, Department of Systems and Control Engineering, University of Malta, Msida, Malta, 2011.
- [142] MALLICK, Satya. *Head pose estimation using OpenCV and Dlib*. 2016-09-26). <https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib>, 2016.

- [143] WANG, Weiwei, et al. Fast head pose estimation via rotation-adaptive facial landmark detection for video edge computation. *IEEE Access*, 2020, 8: 45023-45032.
- [144] BARROS, Jilliam Maria Diaz, et al. Real-time monocular 6-DoF head pose estimation from salient 2D points. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 121-125.
- [145] LI, Dongxing, et al. Visualization Analysis of Learning Attention Based on Single-image PnP Head Pose Estimation. In: 2017 2nd International Conference on Education, Sports, Arts and Management Engineering (ICESAME 2017). Atlantis Press, 2017. p. 1508-1512.
- [146] YE, Mu, et al. Driver Fatigue Detection Based on Residual Channel Attention Network and Head Pose Estimation. *Applied Sciences*, 2021, 11.19: 9195.
- [147] GUPTA, Aryaman, et al. Nose, eyes and ears: Head pose estimation by locating facial keypoints. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019. p. 1977-1981.
- [148] XIA, Jiahao, et al. Head pose estimation in the wild assisted by facial landmarks based on convolutional neural networks. *Ieee Access*, 2019, 7: 48470-48483.
- [149] DAPOGNY, Arnaud; BAILLY, Kevin; CORD, Matthieu. Deep Entwined Learning Head Pose and Face Alignment Inside an Attentional Cascade with Doubly-Conditional fusion. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). IEEE, 2020. p. 192-198.
- [150] ABATE, Andrea F., et al. Near real-time three axis head pose estimation without training. *IEEE Access*, 2019, 7: 64256-64265.

- [151] BARRA, Paola, et al. Web-shaped model for head pose estimation: An approach for best exemplar selection. *IEEE Transactions on Image Processing*, 2020, 29: 5457-5468.
- [152] KAZEMI, Vahid; SULLIVAN, Josephine. One millisecond face alignment with an ensemble of regression trees. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014. p. 1867-1874.
- [153] BULAT, Adrian; TZIMIROPOULOS, Georgios. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017. p. 1021-1030.
- [154] YANG, Heng, et al. Face alignment assisted by head pose estimation. *arXiv preprint arXiv:1507.03148*, 2015.
- [155] GUI, Zhenghui; ZHANG, Chao. 3D head pose estimation using non-rigid structure-from-motion and point correspondence. In: *TENCON 2006-2006 IEEE Region 10 Conference*. IEEE, 2006. p. 1-3.
- [156] JIANG, Min, et al. Head pose estimation based on active shape model and relevant vector machine. In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012. p. 1035-1038.
- [157] WU, Yue; GOU, Chao; JI, Qiang. Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 3471-3480.
- [158] LIU, Leyuan, et al. Head Pose Estimation through Keypoints Matching between Reconstructed 3D Face Model and 2D Image. *Sensors*, 2021, 21.5: 1841.
- [159] BARROS, Jilliam Maria Diaz, et al. Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018. p. 2028-2037.

- [160] WU, Cho-Ying; XU, Qiangeng; NEUMANN, Ulrich. Synergy between 3DMM and 3D Landmarks for Accurate 3D Facial Geometry. In: 2021 International Conference on 3D Vision (3DV). IEEE, 2021. p. 453-463.
- [161] ZHU, Xiangyu, et al. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 2017, 41.1: 78-92.
- [162] TU, Xiaoguang, et al. 3d face reconstruction from a single image assisted by 2d face images in the wild. *IEEE Transactions on Multimedia*, 2020, 23: 1160-1172.
- [163] GUO, Jianzhu, et al. Towards fast, accurate and stable 3d dense face alignment. In: *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIX 16*. Springer International Publishing, 2020. p. 152-168.
- [164] RUAN, Zeyu, et al. SADRNet: Self-Aligned Dual Face Regression Networks for Robust 3D Dense Face Alignment and Reconstruction. *IEEE Transactions on Image Processing*, 2021.
- [165] MARTIN, Manuel; VAN DE CAMP, Florian; STIEFELHAGEN, Rainer. Real time head model creation and head pose estimation on consumer depth cameras. In: 2014 2nd International Conference on 3D Vision. IEEE, 2014. p. 641-648.
- [166] MEYER, Gregory P., et al. Robust model-based 3d head pose estimation. In: *Proceedings of the IEEE international conference on computer vision*. 2015. p. 3649-3657.
- [167] STORER, Markus; URSCHLER, Martin; BISCHOF, Horst. 3d-mam: 3d morphable appearance model for efficient fine head pose estimation from still images. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, 2009. p. 192-199.

- [168] YANG, Tsun-Yi, et al. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. p. 1087-1096.
- [169] AHN, Byungtae; PARK, Jaesik; KWEON, In So. Real-time head orientation from a monocular camera using deep neural network. In: Asian conference on computer vision. Springer, Cham, 2014. p. 82-96.
- [170] LIU, Xingyu. Head pose Estimation Using Convolutional Neural Networks. 2016.
- [171] PATACCHIOLA, Massimiliano; CANGELOSI, Angelo. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 2017, 71: 132-143.
- [172] AMADOR, Elvira, et al. Benchmarking head pose estimation in-the-wild. In: Iberoamerican Congress on Pattern Recognition. Springer, Cham, 2017. p. 45-52.
- [173] HSU, Heng-Wei, et al. Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, 2019, 21.4: 1035-1046.
- [174] DAI, Donggen; WONG, Wangkit; CHEN, Zhuojun. RankPose: Learning Generalised Feature with Rank Supervision for Head Pose Estimation. arXiv preprint arXiv:2005.10984, 2020.
- [175] SHEKA, Andrey; SAMUN, Victor. Knowledge Distillation from Ensemble of Offsets for Head Pose Estimation. arXiv preprint arXiv:2108.09183, 2021.
- [176] WANG, Yujia, et al. A deep coarse-to-fine network for head pose estimation from synthetic data. *Pattern Recognition*, 2019, 94: 196-206.
- [177] KUHNKE, Felix; OSTERMANN, Jorn. Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous la-

- bel spaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. p. 10164-10173.
- [178] BASAK, Shubhajit, et al. Learning 3D head pose from synthetic data: A semi-supervised approach. *IEEE Access*, 2021, 9: 37557-37573.
- [179] BERRAL-SOLER, Rafael, et al. RealHePoNet: a robust single-stage ConvNet for head pose estimation in the wild. *Neural Computing and Applications*, 2021, 33.13: 7673-7689.
- [180] DHINGRA, Naina. LwPosr: Lightweight Efficient Fine Grained Head Pose Estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022. p. 1495-1505.
- [181] PAPAZOV, Chavdar; MARKS, Tim K.; JONES, Michael. Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 4722-4730.
- [182] SAEED, Anwar; AL-HAMADI, Ayoub. Boosted human head pose estimation using kinect camera. In: 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015. p. 1752-1756.
- [183] DROUARD, Vincent, et al. Head pose estimation via probabilistic high-dimensional regression. In: 2015 IEEE international conference on image processing (ICIP). IEEE, 2015. p. 4624-4628.
- [184] GOU, Chao, et al. Coupled cascade regression for simultaneous facial landmark detection and head pose estimation. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 2906-2910.
- [185] YU, Yu; MORA, Kenneth Alberto Funes; ODOBEZ, Jean-Marc. Robust and accurate 3D head pose estimation through 3DMM and online head model reconstruction. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). Ieee, 2017. p. 711-718.

- [186] CANTARINI, Giorgio, et al. HHP-Net: A light Heteroscedastic neural network for Head Pose estimation with uncertainty. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022. p. 3521-3530.
- [187] LATHUILIERE, Stephane, et al. Deep mixture of linear inverse regressions applied to head-pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. p. 4817-4825.
- [188] ZHANG, Hao, et al. FDN: Feature decoupling network for head pose estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020. p. 12789-12796.
- [189] WANG, Haofan; CHEN, Zhenghua; ZHOU, Yi. Hybrid coarse-fine classification for head pose estimation. arXiv preprint arXiv:1901.06778, 2019.
- [190] XIN, Miao; MO, Shentong; LIN, Yuanze. EVA-GCN: Head Pose Estimation Based on Graph Convolutional Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. p. 1462-1471.
- [191] BERG, Axel; OSKARSSON, Magnus; O'CONNOR, Mark. Deep ordinal regression with label diversity. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021. p. 2740-2747.
- [192] HU, Zhongxu, et al. Deep convolutional neural network-based Bernoulli heatmap for head pose estimation. *Neurocomputing*, 2021, 436: 198-209.
- [193] DHINGRA, Naina. HeadPosr: End-to-end Trainable Head Pose Estimation using Transformer Encoders. In: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021). IEEE, 2021. p. 1-8.
- [194] GAO, Bin-Bin, et al. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 2017, 26.6: 2825-2838.

- [195] ZHANG, Wei, et al. Cross-cascading regression for simultaneous head pose estimation and facial landmark detection. In: Chinese Conference on Biometric Recognition. Springer, Cham, 2018. p. 148-156.
- [196] LIU, Zhaoxiang, et al. Facial pose estimation by deep learning from label distributions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019. p. 0-0.
- [197] SHAO, Mingzhen, et al. Improving head pose estimation with a combined loss and bounding box margin adjustment. In: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019). IEEE, 2019. p. 1-5.
- [198] XU, Luhui; CHEN, Jingying; GAN, Yanling. Head pose estimation with soft labels using regularized convolutional neural network. Neurocomputing, 2019, 337: 339-353.
- [199] XUE, Aoru, et al. Robust landmark-free head pose estimation by learning to crop and background augmentation. IET Image Processing, 2020, 14.11: 2553-2560.
- [200] ZHANG, Kaipeng, et al. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 2016, 23.10: 1499-1503.
- [201] VIOLA, Paul; JONES, Michael J. Robust real-time face detection. International journal of computer vision, 2004, 57.2: 137-154.
- [202] JIANG, Huaizu; LEARNED-MILLER, Erik. Face detection with the faster R-CNN. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). IEEE, 2017. p. 650-657.
- [203] CHEN, Dong, et al. Joint cascade face detection and alignment. In: European conference on computer vision. Springer, Cham, 2014. p. 109-122.

-
- [204] CAO, Zhe, et al. Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 7291-7299.
- [205] REDMON, Joseph, et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 779-788.
- [206] HU, Peiyun; RAMANAN, Deva. Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 951-959.
- [207] LIU, Wei, et al. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, Cham, 2016. p. 21-37.
- [208] BAI, Jue, et al. A Study of General Data Improvement for Large-Angle Head Pose Estimation. In: International Conference on Computer Analysis of Images and Patterns. Springer, Cham, 2021. p. 199-209.

Appendix A

Links to datasets

Database	Link
300W-LP [4]	https://www.tensorflow.org/datasets/catalog/the300w_lp
AFLW [23]	https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/aflw/
AFLW2000-3D [4]	https://www.tensorflow.org/datasets/catalog/aflw2k3d
AFW [22]	https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/
AISL [18]	http://www.aisl.cs.tut.ac.jp/dataset_head_orientation.html
AutoPOSE [33]	https://autopose.dfki.de/
BioVid Heat Pain [36]	https://www.iikt.ovgu.de/BioVid.html
BIWI Kinect [5]	https://www.kaggle.com/kmader/biwi-kinect-head-pose-database
BJUT-3D [26]	http://www.bjpu.edu.cn/sci/multimedia/mul-lab/3dface/facedatabase.htm
Bosphorus [25]	http://bosphorus.ee.boun.edu.tr/default.aspx
BU [30]	https://www.cs.bu.edu/groups/ivc/HeadTracking/Home.html
CAS-PEAL [19]	http://www.jdl.ac.cn/peal
CAVE [38]	https://www.cs.columbia.edu/CAVE/databases/columbia_gaze/
CCNU [15]	
CMU Panoptic [8]	<i>Original database:</i> http://domedb.perception.cs.cmu.edu/ <i>Database processed for head pose:</i> https://github.com/Ascend-Research/HeadPoseEstimation-WHENet/issues/13
Dali3DHP [17]	
DD-Pose [32]	https://dd-pose-dataset.tudelft.nl/eval/
DriveAHead [35]	https://cvhci.anthropomatik.kit.edu/~aschwarz/driveahead/
ETH [27]	https://data.vision.ee.ethz.ch/cvl/vision2/datasets/headposeCVPR08/
FacePix [28]	https://cubic.asu.edu/content/facepix-database
GI4E-HP [14]	http://www.unavarra.es/gi4e/databases?languageId=1

GOTCHA-I [42]	https://gotchaproject.github.io/
ICT-3DHP [21]	http://multicomp.cs.cmu.edu/resources/ict-3d-headpose-database-2/
IDIAP-HP [31]	https://www.idiap.ch/en/dataset/headpose
M2FPA [9]	https://pp2li.github.io/M2FPA-dataset/
McGill [20]	https://sites.google.com/site/meltemdemirkus/mcgill-unconstrained-face-video-database
MDM corpus [34]	https://ecs.utdallas.edu/research/researchlabs/msp-lab/MDM.html
Multi-Pie [24]	https://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html
MTFL [?]	http://mmlab.ie.cuhk.edu.hk/projects/TCDCN.html
Pandora [7]	https://aimagelab.ing.unimore.it/pandora/
PIE [39]	https://www.ri.cmu.edu/project/pie-database/
Pointing'04 [29]	http://crowley-coutaz.fr/Head%20Pose%20Image%20Database.html
SASE [12]	https://icv.tuit.ut.ee/databases/
SyLaHP	https://www.iikt.ovgu.de/LmHeadPoseEstBench.html
SynHead [11]	http://www.tnt.uni-hannover.de/papers/view_paper.php?id=1419
Synthetic [16]	https://liangwei-bit.github.io/web/project/icip16_headpose/
Taiwan RoboticsLab [40]	http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm
UbiPose [10]	https://www.idiap.ch/en/dataset/ubipose
UET-Headpose [2]	
UMDFace [6]	http://umdfaces.io/
VGGFace2 [3]	https://github.com/ox-vgg/vgg_face2

Table A.1: Links to available datasets for head pose estimation