## 0.1 Face Recognition

Face Recognition (FR) is a thoroughly debated and extensively researched task in the Computer Vision community for more than two decades [44], popularized in the early 1990s with the introduction of the Eigenfaces [51] or Fisherfaces [42] approaches. These methods projected faces in a low-dimensional subspace assuming certain distributions, but lacked the ability to handle uncontrolled facial changes that broke said assumptions, henceforth, bringing about face recognition approaches through local-features [15, 2] that, even though, presented considerable results, weren't distinctive or compact. Beginning in 2010, methods based on learnable filters arose [65, 33], but unfortunately revealed limitations when nonlinear variations were at stake.

Earlier methods for FR worked appropriately when the data was handpicked or generated on a constrained environment, however, they didn't scale adequately in the real world were there are large fluctuations in, particularly, pose, age, illumination, background scenario, the presence of facial occlusion [44] and many unimaginable more. These shortcomings can be dealt with by using Deep Learning, a framework of techniques that solves the nonlinear inseparable classes problem ref., more specifically a structure called Convolutional Neural Network (CNN) [55].

CNNs are an Artificial Neural Network (ANN) that exhibit a better performance on image or video-based tasks compared to other methods [32]. They were greatly hailed in 2012, after the AlexNet [29] victory, by a great margin, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Just two years later, DeepFace [47] revolutionized the benchmarks scores by achieving state-of-the-art results that approached human performance, reinforcing even further the importance of Deep Learning and shifting the research path to be taken [55].

Given what has been stated so far and the proven robustness, performance, and overall results in computer vision ref. won competitions, the methods discussed in this dissertation will therefore deal exclusively with Deep Learning approaches. For more information on other methods, please refer to [30].

## 0.2 A Face Recognition System

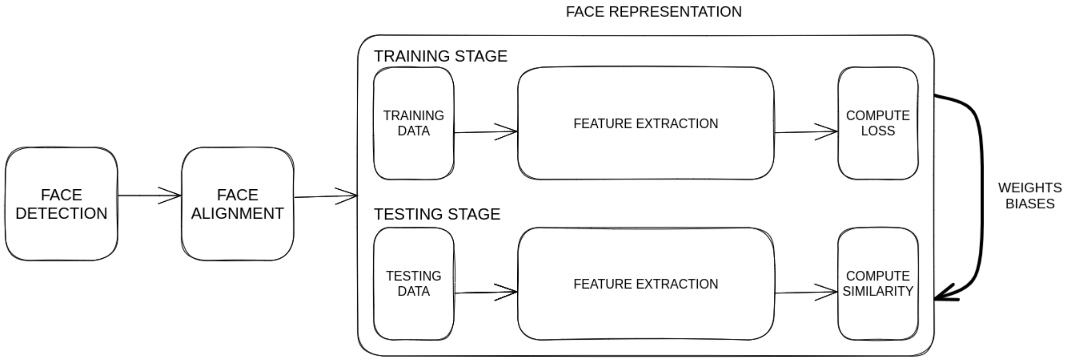


Figure 1: A typical face recognition pipeline, guided by the approach in [55].

According to Ranjan *et al.* [44], the goal of a FR system is to find, process and learn from a face, gathering as much information as possible, and as a result, it is one of the most widely implemented biometric system solutions in light of its versatility when facing real world application [20], such as military, public security and daily life.

By and large, all end-to-end automatic face recognition systems follow a sequential and modular[1] pipeline (Figure 1) composed of three pillar stages [55]: face detection, face alignment and face representation. First an image or video feed is used as an input then, as the name suggests, the **face detection** module is responsible for finding a face. Next, the **face alignment** phase applies spatial transformations to the data in order to normalize the faces' pictures (or frames, in the case where a video is used) to a standardized view. Finally, the **face representation** stage, makes use of deep learning techniques to learn discriminative features that will allow the recognition.

All three stages have their individual importance and methods of implementation[2]. **Face detection** is achievable through classical approaches [53, 8] or deep methods, among them is [19] and the widely applied [70]. **Face alignment**, once again, can be accomplished through traditional measures [16, 39] or more modern ones, namely [27] or the aforementioned [70] which concurrently performs detection and alignment. To conclude, the **face representation** module is no exception,

[1] Sequential because each stage relies on the output from the previous ones, and modular in the sense that each stage employs its own method and it can be modified to better adapt to specific tasks.

[2] For a deeper and extensive study, please refer to: [66] in the case of classic face detection approaches and [40] for deep learning based methods; [57] addresses traditional face alignment methods and is complemented with [20] for more up-to-date techniques; and [30] tackles classic face representation (add the following if needed) while X supplements the deep learning ones.

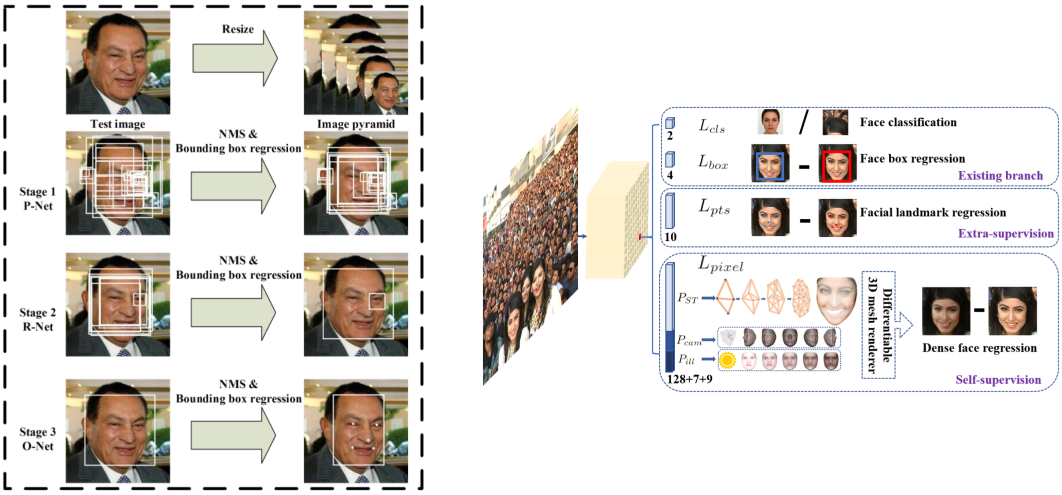
and can also be divided in two groups, regarding the methodology used. Some conventional systems were already mentioned, such as [42, 51], and the deep learning ones are the object of discussion of this dissertation and will be reviewed along the following sections, therefore, the focus will be on describing, with particular interest, the face representation stage.

### 0.2.1 Face Detection

Face detection is the first step in any automatic facial recognition system. Given an input image to a face detector module, it is in charge of detecting every face in the picture and returning bounding-boxes coordinates, for each one, with a certain confidence score [20, 44].

Previously employed traditional face detectors cite here are incapable of detecting facial information when faced with challenges such as variants in image resolution, age, pose, illumination, race, occlusions or accessories (masks, glasses, makeup) [20, 44]. The progress in deep learning and increasing GPU power led DCNNs to become a viable and reliable option that solves said problems in face detection.

These techniques can be included in different categories. A more analytical perspective [20] distributes the methods, depending upon their architecture or purpose of application, over seven categories: multi-stage, single-stage, anchor-based, anchor-free, multi-task learning, CPU real-time and, finally, problem-oriented. Additionally, being as the face detection problem can be seen as a specific task in a general object detection situation, it is no surprise that several works inherit from them and, therefore, some bases are referenced throughout the next list.



a)

b)

Figure 2: Comparison between **a)** MTCNN: multi-stage, CPU real-time and multi-task learning, and **b)** RetinaFace: single-stage, anchor-based, CPU real-time and multi-task learning. MTCNN [70] proposes a series of bounding boxes then, through a series of refinement stages, the best solution and landmarks are found. RetinaFace [19] accomplishes, in a single-stage, face classification and bounding box regression by evaluating anchors, landmark localization and dense 3D projection for facial correspondence.

→ **Multi-stage** methods [19] include all the coarse-to-fine facial detectors that work in similar manner to the following two phases. First, bounding box proposals are generated by sliding a window through the input. Then, over one or several subsequent stages, false positives are rejected and the approved bounding boxes are refined. To complement, one widely applied object detection protocol that inspired face detection methods and perfectly describes the steps mentioned above is Faster R-CNN [45]. However, these methods can be slower and have a more complex way of training [62].

→ **Single-stage** approaches [19] are the ones that perform classification and bounding box regression without the necessity of a proposal stage, producing highly dense face locations and scales. This structure takes inspiration, once again, from general object detectors, for example, the Single Shot MultiBox detector, commonly referred to as SSD [37]. Finally, the methods included in this class are more efficient, but can incur in compromised accuracy, when compared to multi-stage.

→ **Anchor-based** techniques [38, 19, 68] detect faces by predefining anchors with different settings (scales, strides, number, etc.) on the feature maps, then performing classification and bounding box regression on them until an acceptable output is found. As proven by Liu and Tang *et al.* [38], the choice of anchors highly influences the results of prediction. Hence, it is necessary to fine-tune them on a situation-by-situation basis, otherwise, there is a limitation in generalization. Furthermore, higher densities of anchors directly generate an increase in computational overhead.

→ **Anchor-free** procedures, obviously, do not need predefined anchors in order to find faces. Alternatively, these methods address the face detection by using different techniques. For example, DenseBox [26] which attempts to predict faces by processing each pixel as a bounding box, or CenterFace [62] that treats face detection as a key-point estimation problem by predicting the center of the face and bounding boxes. Even so, relating to the accuracy of anchor-free approaches, there's still room for improvement for false positives and stability in the training stage [20].

→ **Multi-task learning** are all the methodologies that conjointly performs other tasks, namely facial landmark[3] localization, during face classification and bounding box regression [20]. CenterFace [62] is one example, and so it is the widely implemented MTCNN [70], which correlated bounding boxes and face landmarks. RetinaFace [19] is another state-of-the-art approach, it mutually detects faces, respective landmarks and performs dense 3D face regression.

→ **CPU real-time** methods, as the name suggests, include the detectors that can run on a single CPU core, in real-time, for VGA-resolution input images. A face detector can achieve great results in terms of accuracy, but for real world applications, its use can be too computational heavy, therefore, can't be deployed in real time (specially in devices that do not have a GPU) [20]. MTCNN [70], Faceboxes [71], CenterFace [62] or RetinaFace [19] are examples of this category.

→ **Problem-oriented** is a category that includes the detectors that are projected to resolve a wide range of specific problems, for example, faces that are tiny, partially occluded, blurred or scale-invariant face detection [20]. PyramidBox [49] is an example that solves the partial occluded and blurry faces, and HR [25] tackles the tiny faces challenge.

Although this distribution can create some overlap among the categories, it is superior due to the simplicity of inferring what defines each category and being a more fine-grained way of classifying techniques when compared to others, namely the dual categorical division by [44] that groups the methods in region[4] or sliding-window[5] based.

### 0.2.2 Face Alignment

Face Alignment, or facial landmark detection [12], is the second stage of the face recognition pipeline, and has the objective of calibrating the detected face to a canonical layout, through landmark-based or landmark-free approaches, in order to leverage the core final stage of face representation [20].

Despite the fact that traditional face alignment methods are very accurate, that only occurs in constrained circumstances. Therefore, once again, to address that issue, deep learning-based methods are the solution to perform an accurate facial landmark localization that realistically scales to real world scenarios [21].

[3] A facial landmark is a key-point in a face that contributes with important geometric information, namely the eyes, nose, mouth, etc. [21]

[4] Region-based approaches creates thousands of generic object-proposals for every image, and subsequently, a DCNN classifies if a face is present in any of them.

[5] Sliding-window approaches centers on using a DCNN to compute a face detection score and bounding box at every location in a feature map.

Furthermore, face alignment, can be accomplished through two categories of methods: landmark-based and landmark-free.

→ **Landmark-based alignment** is a category of methods that exploits the facial landmarks with the aim of, through spatial transformations, calibrating the face to an established layout [20]. This can be accomplished through: coordinate regression, heatmap regression or 3D Model Fitting. **Coordinate regression-based** methodologies [21, 36, 70] consider the landmark localization as a numerical objective, i.e. a regression, thus an image is fed to a DCNN and it will output a vector of landmark coordinates. **Heatmap Regression** [18, 59, 13] is different from coordinate regression because, although it is a numerical objective task, the output is not a coordinate vector, but a map of likelihood of landmarks' locations. Finally, **3D Model Fitting** [7, 12, 61]is the category that integrates methods that consider the relation between 2D facial landmarks and the 3D shape of a generic face. The particularity of them is the reconstruction of the 3D face from a 2D face image that is then projected over a plane in order to obtain the landmarks.

→ **Landmark-free alignment**, on the other hand, integrates the approaches that do not rely on landmarks as a reference to align the face, in contrast, these type of methods incorporate the alignment into a DCNN that gives, as a result, an aligned face [20]. An example of an end-to-end method that does not depend on facial landmarks is RDCFace [72], and it rectifies distortions, applies alignment transformations and executes face representation. Hayat et al. [24] proposes a method that deals with extreme head poses. The process to register faces in an image with high pose variance can be quite challenging and often demands complex pre-processing, namely landmark localization, therefore, to address that, a DCNN is employed that does not rely on landmark localization and concomitantly register and represent faces.

As can be seen from the previous section, this step in the face recognition process can be accomplished, very sporadically, through standalone methods that process the detected face from the previous stage, but generally joint detection and alignment methods (and sometimes even face representation), previously referenced in the multi-task learning definition, are the optimal choice [12].

### 0.2.3 Face Representation

Finally, Face Representation is the last stage of the Face Recognition process. It is responsible for processing the aligned face from the previous stage and mapping the produced feature representation to a feature space, in which features from the same person are closer together and those that are different stand further apart from each other [20].

According to the literature [20, 34, 44, 46, 55], there's a consensus about how Face Recognition can be performed in two settings of operation: face verification and face identification. This distinction is only made possible due to the approaches available in the Face Representation stage that can leverage one, the other or both.

→ **Face verification**, also referred to as **face authentication**, is a one-to-one match, and it's the action of verifying if the query face matches the identity that's being claimed. These principles are used in biometric systems such as self-service immigration clearance using E-passport. [34]

→ **Face identification**, also called **face recognition**, is a one-to-many correlation process that compares a query face to a database of faces and associates it to the corresponding match (or matches). A typical use case is to identify someone in a watchlist or surveillance videos. [34]

The overall pipeline comes to a conclusion in this module, however, in reality, it goes further than that. As can be seen in (Figure 1), due to its importance for the face recognition problem, it's highlighted the inherent pipeline of the Face Representation stage, henceforth, it shall be discussed in depth in the next section.

## 0.3 Face Representation Pipeline

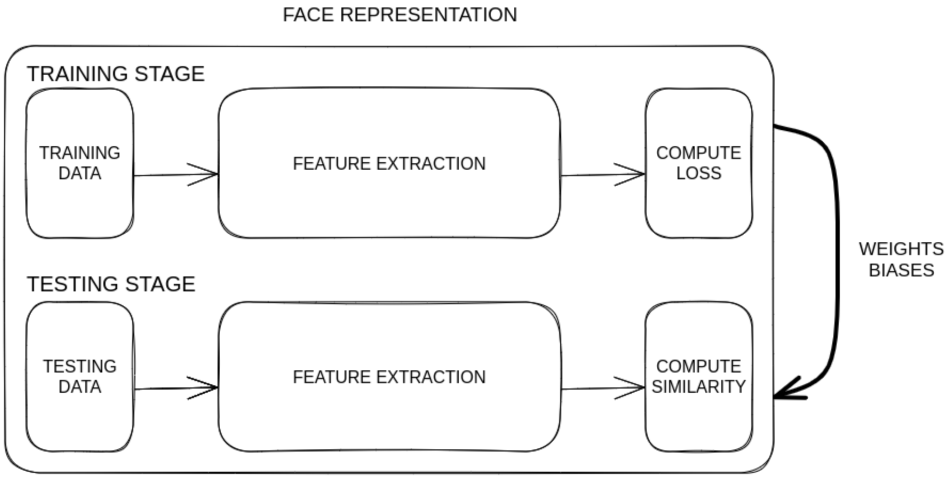


Figure 3: Face Representation pipeline, guided by the approach in [55].

As show in (Figure 3), Face Representation is a two-step module composed of a training and testing stage. So as to be capable of performing face recognition, in either a verification or identification manner, a face representation system needs to learn robust, invariant and discriminative features that can distinguish identities [44].

To meet these requirements, the feature extractor must first be trained properly by taking data from previous stages and outputting a feature representation that's compared to the desired value using a loss function[6] [31, 55]. After that, everything is ready for the testing stage, where the face recognition *per se* occurs by calculating a similarity score for the feature representation produced by the trained feature extractor, and dictating if the identity belongs to the same person (face verification) or if it matches any identity (face identification) [44].

### 0.3.1 Datasets: Training and Testing Data

As been discussed throughout this dissertation, Deep Learning techniques can solve the problem of handling unconstrained scenarios, where there are variations in pose, illumination, occlusion, and so forth. To support this, in the past few years, datasets have been developed with this in mind so to be able to provide a large and diverse set of both training data, allowing for adequate regularization to unseen circumstances, and testing data that benchmarks the face recognition system in, as similar as possible, unconstrained real world scenarios [20].

[6] Also referred to as a cost or objective function. [31]

## Training Data

When developing a deep face recognition system it's essential to keep in mind its necessity to adapt, and that's where the dataset used for training comes at play. Large training datasets are essential for face recognition [43], but large-scale is not enough. There must be a balance between the depth (number of unique identities) and the breadth/width (number of images per identity) [6, 10], and it will lead to different effects.

On one hand, a training dataset that is deep will help the face recognition system to produce more discriminative feature representations, since it will have a great number of identities to learn from. On the other hand, a wider set will have more images per identity, therefore, variations in pose, expressions, illuminations, occlusions, background clutter, image quality, accessories, and so forth [5] can be introduced and ultimately lead to feature representations more robust to them.

In the following pages, a mix of training datasets, of overall relevance and more geared towards the purpose of this dissertation, will be reviewed.

→ **CASIA-WebFace** [64], composed of 494,414 face images and 10,575 identities, was proposed as a novel dataset to overcome the problem of data dependence in face recognition and improve comparability across different methods. By training on the same dataset, methods can be better evaluated and compared.

→ **VGGFace** [43] was published alongside a homonymous face recognition method and, once again, with the objective of combating the lack of available large scale public datasets. It contains 2,6 million images and 2,622 different identities and a curated version, where incorrect image labels were hand-removed by humans, has 800,000 images for the same amount of identities.

→ **MS-Celeb-1M**'s [23] first intention was to provide a novel benchmark to identify celebrities that solves name ambiguities by linking a face with an entity key in a knowledge base. Second, it aimed at solving the gap in available large-scale datasets by providing a training set with, approximately, 10 million images and 100 thousand identities. Unfortunately, it is a dataset known for the presence of noisy labels.

→ **MegaFace** [41] introduced a benchmark for million-scale face recognition and provided a public large-scale training dataset that integrated 4,753,320 faces over 672,057 identities. The main difference compared to the previously mentioned datasets is that MegaFace does not use celebrities as subjects, in contrast it leverages the photographs released by Flickr under the Creative Commons license.

→ **VGGFace2** [10] is another large-scale dataset, and its main goals are: 1) covering numerous identities, 2) reduce labeling noise through automatic and manual filtering and, finally, 3) represent more realistic unconstrained scenarios due to a novel dataset generation pipeline that gathers images with a broad range of poses, age, illumination and ethnicity. All in all, this resulted in a dataset comprised of 3,31 million faces of 9131 subjects.

→ **UMDFaces-Videos** [6] is a video-based dataset composed of 22,075 videos of 3,107 subjects with 3,735,476 human annotated frames with great variation in image quality, pose, expressions and lightning. It was proposed during a study how the performance of a face verification models is impacted by the effects of: 1) the type of media used for training (only videos or still images vs a mixture of both), 2) the width and depth of a dataset, 3) the label's noise and 4) the alignment of the faces.

→ **Celeb-500k** [9] is another large-scale proposed with two issues in mind: the disparity in the scale of public datasets when compared with private ones, and determining the impact in performance from intra- and inter-class variations. That being so, Celeb-500k, consisting of 50 million images from 500 thousand persons, and Celeb-500k-2R, a cleaned version of the previous, comprised of 25 million aligned faces of 245 thousand identities, are released.

→ **IMDb-Face** [54] proposes a new dataset with based on a manually cleaned revision of MS-Celeb-1M and MegaFace. The growing demand for large-scale datasets introduced a new variable to take into consideration: the time available to annotate the data. Datasets that are well-annotated and have an enormous amount of data are notably expensive and time-consuming to develop. Therefore, automatic measures to clean the data were used, so it's expected for a certain degree of noise to be introduced in a dataset. After selecting a subset from both the originals datasets, 2 million images were manually cleaned and resulted in 1,7 million images of 59 thousand celebrities.

→ **MS1MV2** [17] is another well know dataset. It was proposed in the ArcFace face recognition method paper and consists of a semi-automatic refinement of the previously mentioned MS-Celeb-1M, resulting in 5,8 million images of 85 thousand identities.

→ **RMFRD** [58] is presented in the context of the need of using a mask, mandated by the COVID-19 pandemic, and that greatly reduces the effectiveness of conventional face recognition methods. Therefore, there's was a need to improve their performance and for that a dataset that provides masked faces is needed. RMFRD pioneered this need by publishing a dataset consisting of 5 thousand masked and 90 thousand unmasked faces from 525 celebrities.

→ **Glint360K** [4] is a training set presented in the Partial FC method paper. It was generated by merging and cleaning the aforementioned Celeb-500K and MS1MV2 datasets, which resulted in 17 million images of 360 thousand individuals.

→ **WebFace260M** [75] takes a giant leap in closing the gap between public available datasets and private ones. Partnered with a time-constrained face recognition protocol, the original paper presented an enormous 260 million faces and 4 million identities noisy dataset, an automatically cleaned, high quality training set with 42 million faces over 2 million identities (WebFace42M), and a smaller scale training dataset derived from the WebFace42M that has 10% of its data (WebFace4M).

→ **DigiFace-1M** [5] is a novel approach that revolutionizes the way of training face recognition models. It is a fully synthetic dataset that proposes mitigating three very relevant problems present in the majority of the conventional datasets: 1) ethical issues, 2) label noise and 3) data bias. The dataset is divided in two parts: part one contains 720 thousand images from 10 thousand identities and part two has 500 thousand images with 100 thousand identities, for a total of 1,22 million images and 110 thousand unique identities.

| **Dataset** | **Year** | **Availability** | **Images/videos** | **Depth** | **Avg. Breadth** | **Distribution** | **Description** |
|---|---|---|---|---|---|---|---|
| CASIA-WebFace [64] | 2014 | Discontinued | 494,414/- | 10,575 | 46.7 | Public | First public face recognition dataset. |
| Facebook [48] | 2015 | - | 500M/- | 10M | 50 | Private | temp |
| Google [46] | 2015 | - | 200M/- | 8M | 25 | Private | temp |
| VGGFace [43] | 2015 | Discontinued | 2,6M/- | 2,622 | 991.6 | Public | High width public dataset released alongside VGGFace method |
| MS-Celeb-1M [23] | 2016 | Discontinued | 10M/- | 100k | 100 | Public | Large-scale celebrities dataset. |
| MegaFace [41] | 2016 | temp | 4,753,320/- | 672,057 | 7.1 | Public | Non-celebrity dataset |
| VGGFace2 [10] | 2017 | temp | 3,31M/- | 9131 | 362.5 | Public | High characteristics variation dataset |
| UMDFaces-Videos [6] | 2017 | temp | -/22,075 | 3,107 | 7.1 | Public | Video-based dataset with great variations |
| Celeb-500k [9] | 2018 | temp | 50M/- | 500k | 100 | Public | Noisy celebrities dataset |
| Celeb-500k-2R [9] | 2018 | temp | 25M/- | 245k | 102 | Public | Cleaned version. |
| IMDb-Face [54] | 2018 | temp | 1,7M/- | 59k | 28.8 | Public | Manually cleaned revision of MS-Celeb-1M and MegaFace. |
| MS1MV2 [17] | 2019 | temp | 5,8M/- | 85k | 68.2 | Public | Semi-automatic cleaned version of MS-Celeb-1M |
| RMFRD [58] | 2020 | temp | 95k/- | 525 | 180.9 | Public | temp |
| Glint360k [4] | 2021 | temp | 17M/- | 360k | 47.2 | Public | temp |
| WebFace260M [75] | 2021 | temp | 260M/- | 4M | 65 | Public | temp |
| WebFace42M [75] | 2021 | temp | 42M/- | 2M | 21 | Public | temp |
| WebFace4M [75] | 2021 | temp | 4,2M/- | 200k | 21 | Public | temp |
| DigiFace-1M [5] | 2022 | temp | 1,22M/- | 110k | 11.1 | Public | temp |

Table 1

## Testing Data

After the training is completed the performance of the system needs to be evaluated on different challenges to properly access if it scales (or generalizes or applies or performs in) to real-world scenarios. A test dataset can be chosen for specific hurdles, for instance, cross-pose, cross-age, racial variations, quality assessment, and so forth [20].

→ **QMUL-SurvFace** [14] is a dataset introduced as a benchmark in the *Surveillance Face recognition Challenge* for face recognition in a surveillance context. By data-mining 17 public person re-identification datasets, it achieves 463,507 facial images of 15,573 identities collected in uncooperative surveillance scenarios. Consequently, it presents a high variance in resolution, motion blur, pose, occlusion, illumination and background clutter.

→ **LFW**

→ **YTF**

→ **IJB-C**

→ **MegaFace**

→ **Trillion Pairs**

→ **DiF**

→ **Fair Face**

→ **RFW**

→ **MDMFR** [52] is brought about in light of the COVID-19 impact, where wearing a mask became mandatory. Unfortunately, this rendered unusable the traditional face recognition methods because they were designed to recognize unveiled faces. Therefore, in conjunction with Deepmasknet, a framework that performs both masked detection and recognition, MDMFR was released

→ **CAFR** (rev 2021) [73]

→ **XQLFW** (2022)

→ **FaVCI2D** (2022)

Although some of the previously referenced datasets do have a benchmarking component and are described as such, they can also be generally employed to train algorithms. When it is desired to finetune[7] a model for a specific challenge that, usually, can be better achieved by employing a ”benchmark” dataset that was purposefully developed to evaluate a model’s performance on that exact challenge.

> A common denominator throughout the machine learning domain, and more specifically deep learning, is a general difficulty in separating concepts with a single line. This is a non-linear science in its nature.

### 0.3.2 Feature Extraction

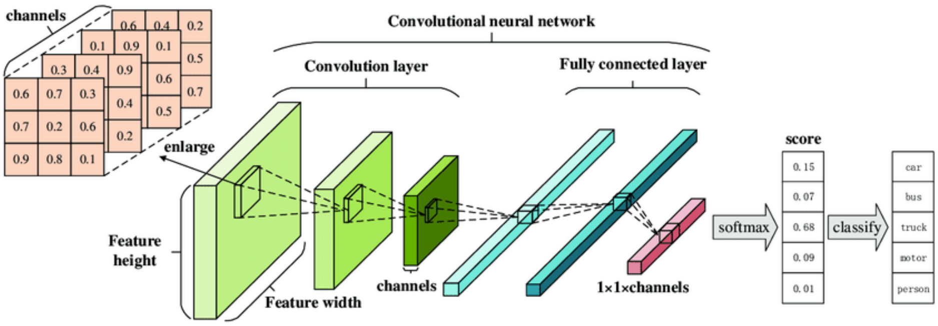


Figure 4: Architecture of a Convolutional Neural Network [28].

There are several types of Neural Networks architectures, but Convolutional Neural Networks (CNNs or Convnets) are probably the most widely implemented model overall [63, 35] with successful applications in the domains of Computer Vision [29, 47, 50, 69] or Natural Language Processing[1, 56, 60]. In the CNN category itself there are different variants, but they all abide the fundamental structure of a feedforward hierarchical multi-layer network (Figure 4). Feedforward because the information only flows in a singular direction without cycling [67], hierarchical because the higher complexity internal representations are learned from lower ones [31, 74] and multi-layer because it is composed of a series of stages, blocks or layers: the raw data is fed to an input layer, forwarded to a sequence of intercalating convolutional and pooling layers, transmitted to a stage of one or more fully-connected layers [31, 63, 22, 3]. The convolutional layer is designed to extract feature representations by being composed of kernels (or filter banks [31]) that compute feature maps through element-wise product, to which is applied a nonlinear activation function [22, 63]. Next is the pooling layer, that’s responsible for reducing the spatial size of the input data [22] and joining identical features [31]. Finally, the fully connected layers, and their core function is to perform high logic and generate semantic information [22].

[7] Producing a new model for a specific task by freezing most of the model’s layers and training only the last few [76]. More details on this topic in the section Transfer Learning.

Using CNNs for Computer Vision tasks is not an arbitrary choice, but due to the fact that the network design can benefit from the intrinsic characteristics of the input data, consequently performing really well in image related applications [31, 11]. In the first place, images have an array-like structure with numerous elements, namely, each pixel has an assigned value organized in a grid-like manner [63]. In the second place, there's an inherent correlation between local groups of values, which creates distinguishable motifs [31]. Finally, the local values of images are invariant to location, that is, a certain composition should have the same value independently of the spatial location in the picture [31]. Therefore, the following key, unique features potentiate the previously stated efficient performance [11]:

1. Designed to process multidimensional arrays [31];
2. Shared weights between the same features in different locations;
3. Automatically identifies the relevant features without any human supervision, hence, small amounts of preprocessing [3, 35];
4. Local connections (or receptive fields/sparse connectivity) [3];
5. Pooling layers that reduces the spatial size of the input data.

The ensemble of features 2, 4 and 5 enable an invariance of the network to small shifts, distortions and rotations [22, 31], while 2, 3, 4 and 5 helps to reduce the complexity of the model, and as a result training it is easier[22, 35].

#### Transfer Learning

### 0.3.3 Loss