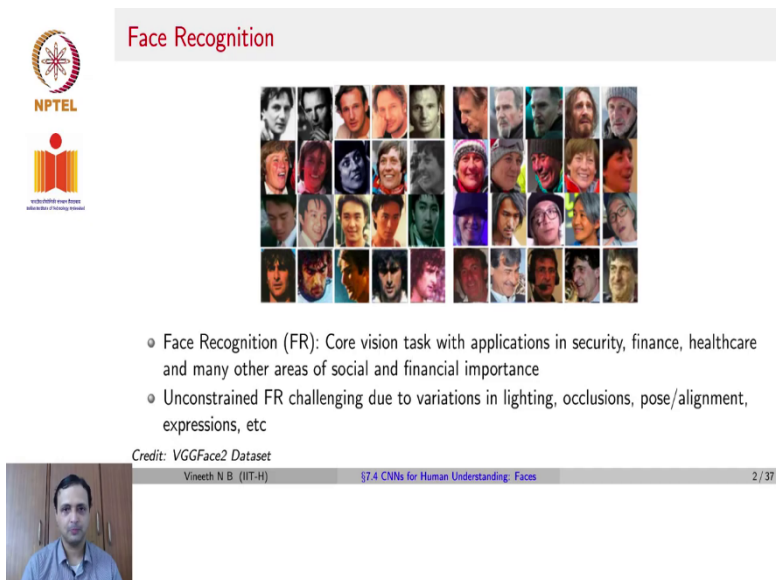


Deep Learning for Computer Vision
Professor. Vineeth N Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
Lecture No. 48
CNN for Human Understanding Faces: Part 01

We will now move to another important task, where CNNs have been extremely useful over the last few years. In computer vision tasks, CNNs are used in understanding humans from various different perspectives. In this first lecture, we look at understanding faces and processing faces for tasks such as recognition and verification.

(Refer Slide Time: 00:46)



The slide features the NPTEL logo on the left and a grid of 30 diverse human faces in the center. Below the grid, there are two bullet points and a credit line.

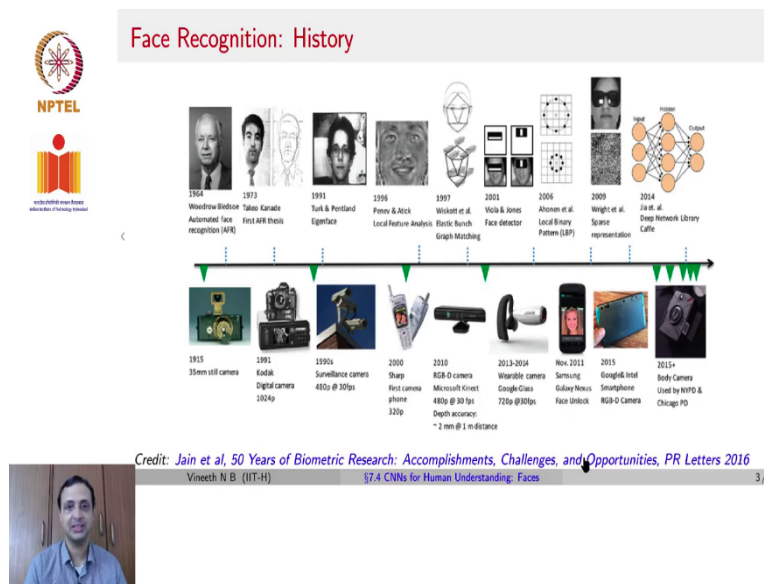
- Face Recognition (FR): Core vision task with applications in security, finance, healthcare and many other areas of social and financial importance
- Unconstrained FR challenging due to variations in lighting, occlusions, pose/alignment, expressions, etc

Credit: VGGFace2 Dataset

Vineeth N.B. (IIT-H) §7.4 CNNs for Human Understanding: Faces 2 / 37

Face Recognition has remained an extremely important computer vision task for several decades now as part of biometrics. It has applications in security, finance, healthcare, and various other aspects of society in finance. Unconstrained face recognition, which is about recognizing faces in the wild, is a very challenging problem, because of the variations that you could have in lighting, in occlusions, in pose or alignment, in expressions, so on and so forth.

(Refer Slide Time: 01:29)



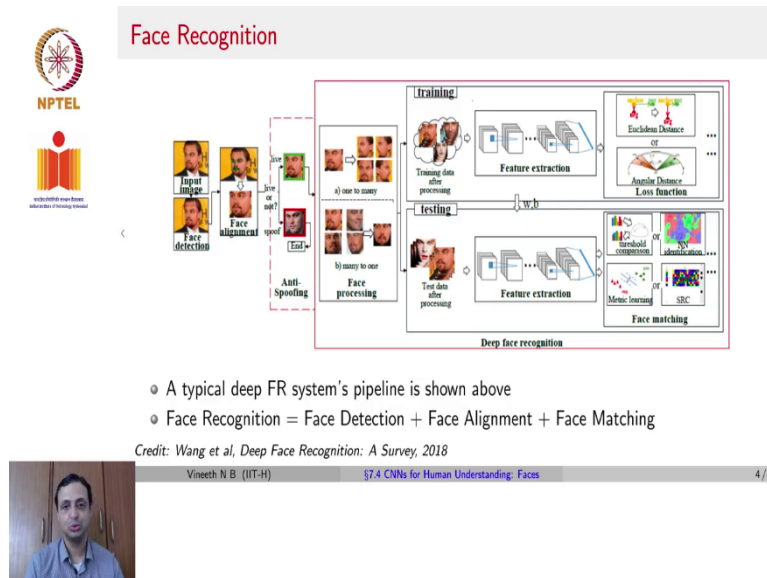
Face recognition is not a new topic, it's been around for several decades now, because of its importance. Algorithmically speaking, the first face recognition efforts started in the 60s with this work (on) from Woodrow Bledsoe. Then came in the 1970s from Takeo Kanade. Then one of the most popular algorithms in early face recognition in the 90s known as Eigenfaces, which repurposes principal component analysis in a slightly different way to perform face recognition. Then came various approaches using Local Feature Analysis for face recognition. Then came Elastic Bunch Graph Matching. Of course, we had the Face Detector from Viola Jones, that was an important component of face recognition systems.

Locally binary patterns, local binary patterns, which we talked about. And we talked about handcrafted features were extensively used for face analysis tasks. Then, towards the end of the first decade of the Twenty First Century, came Sparse Feature Representations for face analysis tasks. And of course, since 2014, Deep Neural Network-based Approaches for face recognition, which we will focus on in this lecture. From a hardware perspective, cameras began way back in 1915 with a digital camera coming in the 90s. Then face recognition shifted to Surveillance Cameras, Camera Smartphones, Kinect-based devices, Microsoft Kinect-based devices that you see on your Xbox, Google Glass kind of devices towards the end of the first decade of Twenty First Century.

And in 2011, Samsung Galaxy had their face unlock feature implemented as part of the smartphone. And then came the RGB-D Camera. And more recently, Body Cameras that can do face recognition. All of this has been very well chronicled in a recent article known as “50

Years of Biometric Research: Accomplishments, Challenges and Opportunities.” Do look at it if you have time.

(Refer Slide Time: 03:51)



A standard face recognition pipeline as used in, as deep learning is employed for, is given by this diagram here. So, you start with an Input Image. You first perform Face Detection, because you have to isolate the faces from the image. Once face detection is done, there are a few pre-processing tasks that need to be done before you give the cropped face for a recognition task. So, the first task is to align the face to a set of predefined geometry. The second task, which is optional depends on the application, is to check whether this image is a spoof or life, “Did I hold up an image to the camera? Or is it really my face?” Once that check is done, if it is spoof, you conclude this pipeline right there.

But if it is not a spoof, we go on to the next stage. The next stage could be looked at as Face Processing. There are two kinds, “one-to-many” and “many-to-one”, once again, depending on what application you want to use these for. We will see both of these in more detail in a couple of slides from now. Of course, the pre-processing task is an optional task, depending on whether you need it for a given application. And once you have your training data, after all of these processing steps, you provide this to a Convolution Neural Network for Feature Extraction.

And at the end, you use different kinds of Loss Functions, such as Euclidean Distance or Angular Distance, so on and so forth to train the CNN. At test time, you have a test image, which goes through a similar round of processing. And once you have a Test Data after

processing, you extract the features, and you finally match the final embedding of that face using various approaches such as matching against the threshold, or doing Nearest Neighbour Matching, or as we will see doing Metric Learning, or Sparse Representations.

We will see some of these over the next few slides. Broadly speaking, one could say that to deploy face recognition systems in the wild, it is a combination of detection, alignment, and matching. As we will see, matching can be of several kinds too when we talk about faces.

(Refer Slide Time: 06:41)

Face Recognition System: Key Components

- **Face Processing**
 - One-to-many Augmentation
 - Many-to-one Normalization
- **Deep Feature Extraction**
 - Network Architecture
 - Loss Function
- **Face Matching** by Deep Features

The diagram illustrates the flow from face processing (one-to-many augmentation and many-to-one normalization) to deep feature extraction (network architecture and loss function), and finally to face matching (feature extraction using deep conv. networks, threshold comparison, identification, metric learning, and SFC).

Credit: Wang et al, Deep Face Recognition: A Survey, 2018




Vineeth N B (IIT-H)

§7.4 CNNs for Human Understanding: Faces

5 / 37



So, the key components are face processing, where you could have a one-to-many augmentation or a many-to-one normalization. We will describe both of these very soon. Then you have a Deep Feature Extraction through a Network Architecture and a corresponding loss function. And finally, you match these embeddings or features or representations using various different approaches.

(Refer Slide Time: 07:14)




Face (Pre-)Processing

- **One-to-many Augmentation**
 - Generating many patches or images of pose variations from a single image (through rotations, for e.g.)
- **Many-to-one Normalization**
 - Attempts to recover canonical view of face images from one or many images of a non-frontal view
 - Can help in preserving identity despite variations in pose, lighting, expression and background



Credit: Wang et al, A Survey on Face Data Augmentation, Neural Computing and Applications, 2019; Qian et al, Unsupervised Face Normalization With Extreme Pose and Expression in the Wild, CVPR 2019

Vineeth N B (IIT-H) §7.4 CNNs for Human Understanding: Faces 6 / 37



What is this pre-processing step that we just spoke about? One can perform one-to-many augmentation, which is the standard data augmentation that we spoke about, where we could generate many patches or many images by varying the face image in different ways. An example could be by rotating the face image in different ways, you could be simulating different poses of the person to an extent. And this kind of an augmentation can help the face recognition system be more robust. On another hand, one may also want to get a single canonical view of a person by normalizing several face images onto a standard model.

For example, you could have all these different images of a single person. As you can see, these have variations from an illumination standpoint, these have variations from a pose standpoint. How do you get these? These could be obtained at different points in time or these could be obtained as frames in a single video sequence, say as a person was speaking, and maybe his head was moving, and you capture all of those frames, and then you normalize all of those frames into a single frame, which gives a frontal view which can be used for further processing.

A lot of work in this space, which we may not be able to focus on in this course, also corresponds to the idea of using 3D models for face recognition. Such a normalization approach can help in preserving identity, despite variations in pose, lighting, expression and background. Because once you normalize for these factors, you are likely to have a canonical image that a CNN may do better on while performing a recognition task.

(Refer Slide Time: 09:24)



Network Architectures for Face Recognition

- Few special CNN architectures proposed for deep FR
- However, most successful backbone networks in deep FR shaped around then SOTA deep object classification networks



Method	Public Time	Loss	Architecture	Number of Networks	Training Set	Accuracy (±SD%)
DeepFace [10]	2014	softmax	AlexNet	1	Facebook (4.0M) UK	97.51 (±0.25)
DeepID [11]	2014	contrastive loss	AlexNet	25	CaltechFaces (0.3M) UK	99.11 (±0.11)
DeepID [11]	2015	contrastive loss	VGGNet-10	50	CaltechFaces (0.3M) UK	99.31 (±0.10)
Facenet [12]	2015	triplet loss	GoogleNet-24	1	Google (500K) UK	99.01 (±0.09)
Ballu [13]	2015	triplet loss	CNN-9	10	Ballu (1.5M) UK	99.37
VGGFace [14]	2015	triplet loss	VGGNet-16	1	VGGFace (1.2M) UK	98.95
SIFTNet [15]	2015	softmax	SIFT-CNN	1	MS-Celeb-1M (1.4M) UK	98.9
Center Loss [16]	2016	center loss	LeNet-7	1	CASIA-WebFace (0.49M) UK	99.28
L-softmax [17]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M) UK	98.71
Range Loss [18]	2016	range loss	VGGNet-16	1	MS-Celeb-1M (1.3M) UK	99.52
L2-softmax [19]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (1.3M) UK	99.78
NormFace [20]	2017	contrastive loss	ResNet-20	1	CASIA-WebFace (0.49M) UK	99.19
CoCo loss [21]	2017	CoCo loss	-	1	MS-Celeb-1M (0.8M) UK	99.86
AMP loss [22]	2017	AMP loss	ResNet-27	1	MS-Celeb-1M (4.0M) UK	99.58
Marginal Loss [23]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (0.8M) UK	99.44
SphereFace [24]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M) UK	99.42
CCF [25]	2018	center moment loss	ResNet-27	1	CASIA-WebFace (0.49M) UK	99.12
AMS loss [26]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M) UK	99.12
ConfNet [27]	2018	confnet	ResNet-64	1	CASIA-WebFace (0.49M) UK	99.33
Anchor [28]	2018	anchor	ResNet-100	1	MS-Celeb-1M (1.3M) UK	99.83
Ring loss [29]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (1.3M) UK	99.59

Credit: Wang et al, Deep Face Recognition: A Survey, 2018



Vineeth N B (IIT-H)

§7.4 CNNs for Human Understanding: Faces

7 / 37

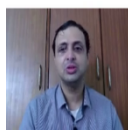
So, what network architectures have been used for face recognition over the years, so we will focus on the deep learning era for face recognition in this particular lecture. At each stage over the last few years, the architecture used for face recognition mirrors the CNN development at that stage. So, if you see this timeline below here, 2012 was AlexNet. So Deep Face, which was one of the first comprehensive efforts of using the new generation CNNs for face recognition, used AlexNet as its main backbone architecture.

Subsequently came Facenet in 2015, that used GoogleNet. VGGface in 2015 that used VGGNet and a network known as SphereFace in 2017, that used ResNet. A network known as VGGFace2 in 2017, that used Squeeze and Excitation net. A more comprehensive listing can be seen here, a table that was obtained from this paper known as Deep Face Recognition: A Survey. And you can see here that the backbone architecture of all of these different methods have predominantly been the popular architectures that have been successful over the last few years. While there have been a few deviations, where researchers have varied the architecture slightly, for a large part, the backbones of face recognition architectures in recent years have been AlexNet, VGG, ResNet, and so on and so forth.

(Refer Slide Time: 11:14)

Face Recognition: Verification and Identification

- Face recognition can be broadly divided into two tasks:
 - Face verification
 - Face identification



Vineeth N B (IIT-H)

§7.4 CNNs for Human Understanding: Faces

8 / 37

The entire space of face recognition can broadly be divided into two kinds of tasks that are required from an application standpoint, verification and identification. In both these cases, generally, it is the last stage of the architecture that changes, you have an image, you perform detection and alignment, you get a cropped image that is well aligned, then you take a deep CNN, a feature extraction network, get a feature representation, which is then passed on to a verification system or an identification system.

(Refer Slide Time: 11:57)

Face Identification

- Assign input image to person name/id from database (one-to-many matching)
- Formulated as K+1 multi-class classification (one additional class for unrecognized faces)
- Input:** Face image
- Output:** Identity class/face ID



Vineeth N B (IIT-H)

§7.4 CNNs for Human Understanding: Faces


9 / 37

What is face identification? Face identification is the task of assigning a given input image to a person name, or identity from a database. It would be a one-to-many matching. So, you have a given image, and you have to match this image with many identities in your database

to find the closest match. So, this task is very similar to the classification task that we have been speaking about so far, such as an ImageNet, where given an image, you match with 1000 different classes, and find which is the class that should be assigned to this image.

This is generally formulated as a $K + 1$ multi-class classification problem, where you do have one additional class, in case the face comes from outside the database. So, your input is the face image and the output is the identity class or the face ID.

(Refer Slide Time: 13:04)

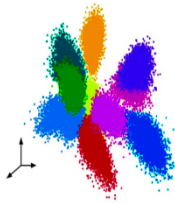


Softmax + CrossEntropy


- Consider last linear layer W, b parameterizes the subjects i.e., maps feature embeddings on to subject labels, cross-entropy loss is

$$\mathcal{L}_{CE} = -\log \left(\frac{e^{W_i x^T + b_i}}{\sum_j e^{W_j x^T + b_j}} \right) \rightarrow \text{logits}$$

- Softmax+CE produces feature embeddings that geometrically looks like ellipsoids i.e., large intra-class variance
- The loss enforces good classification (nice boundaries between classes i.e., small inter-class variance) but it does not enforce small intra-class variance



Source: Maci et al. Deep Face Recognition: A Survey, SIBGRAPI'18. [9]



Vineeth N B. (IIT-H)
§7.4 CNNs for Human Understanding: Faces
10 / 37

What is the loss function used in this scenario? Standard Softmax and CrossEntropy loss that we have been visiting all the while. So, you have a last linear layer, which parameterizes the subjects, the representations of the subjects. And the cross entropy can be given by this form, where the x here is the penultimate layers' representation, on top of which you have some weights, which gives you the final outputs in that last layer, on which you apply the Softmax Activation function, and then the CrossEntropy loss. To remind you here, these values are also known as Logits, the set of outputs of a neural network before you apply the Softmax Activation function are also referred to as the logits of a neural network.

And generally, when you use the Softmax and CrossEntropy loss, these embeddings that you get before you apply the Softmax, geometry look like Ellipsoids, where you have a large intra-class variance because that is not the focus of the cross entropy loss, but you have a good inter-class variance, where you separate these classes using this loss.

(Refer Slide Time: 14:29)

Face Verification

Verification(x, id-1021)

Test Image(x) → Deep CNN $f(x, \theta)$ → f

Face Database: id-1021 (CNN) f_1 , id-1022 (CNN) f_2 , id-1023 (CNN) f_3 , id-1024 (CNN) f_4

Similarity measure → binary decision (0/1)

- Verifying whether two images belong to the same identity
- Ascertain whether image is of claimed person/id (one-to-one matching)
- **Input:** Face image, Face ID
- **Output:** Match/Not match (binary classification)

Vineeth N B (IIT-H) 37.4 CNNs for Human Understanding: Faces 11 / 37

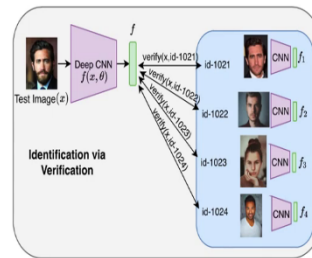
Now for the verification task, the verification task is about verifying whether two images belong to the same identity. This is a one-to-one matching task. Given an input image and an identity that the person claims to have, this task has to ascertain whether the input image belongs to that identity. You could imagine this setting to be in a, in an immigration setting, where a passport is presented to an immigration officer, and the officer looks at your face to check if the face matches the image and the identity on the passport. It is a one-to-one matching. The immigration officer is not comparing your image with a database of identities, but is only verifying whether you are the person you claim to be.

So, in this case, your input is a face image and the face ID unlike the identification setting, and the output is a binary classification problem, which states whether it is a match or not a match. So illustratively speaking, you have an image, you have a convolutional neural network, you get a feature representation, you have your face database from which one identity is picked out based on what the person claims himself or herself to be. And there is a similarity measure that is used to check the similarity between these 2 images, and a binary decision of Match or Not Match is taken.

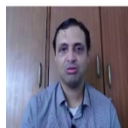
(Refer Slide Time: 16:18)



Identification via Verification



- Identification can be solved via verification approach
- Removes need to retrain model on addition of new face classes to the database \Rightarrow more practical/feasible
- Identification involves multiple verification steps \Rightarrow error gets amplified
- Goal becomes to build an accurate/efficient verification system via learning a robust similarity metric



Vineeth N B (IIT-H)



§7.4 CNNs for Human Understanding: Faces

12 / 37

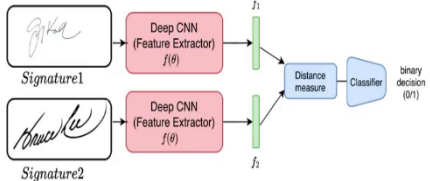
One can also perform identification as a verification task. How is this done? You can now take a test image, take the feature representation through a convolutional neural network, and for each of your identities in your database, you can check whether there is a match or not a match. This would make an identification task as a verification task. Why is this even talked about? Why is this useful? This removes the need to retrain the model on addition of new face classes to the database. So, if you have 1000 people in your database, and you trained a CNN already, if a new person is added, you again have to retrain your CNN, because the number of neurons in your last layer will now get added by 1.

However, if we treat this as a verification problem, we only need the representation of the neural network, and the matching with respect to every new identity can be done using other similarity metrics. This makes this approach scalable. Identification, however, could have multiple verification steps, which means error could get amplified. If there was error anywhere in the pipeline, that could now get amplified, because you are now comparing with multiple verification, that you are going through now, multiple verification steps. In this case, the goal would be to develop an, a very accurate and efficient verification system, so that it can then also be used effectively for identification.

(Refer Slide Time: 18:06)




Verification: Siamese Networks¹



- First proposed for signature verification in 1994
- Two replicas of same architecture parametrized with same weights working in tandem on different inputs
- Network parameters learned via some form of distance measure to extract distinctive features - an idea that is used even today

¹Bromley et al, Signature Verification using a Siamese Time Delay Neural Network, NIPS 1994

Vineeth N B. (IIT-H) 37.4 CNNs for Human Understanding: Faces 13 / 37



One of the earliest efforts for verification was the Siamese Networks way back in 1994. This was first proposed for the task of signature verification. So, a certain bank has a person's signature on their records. And if a person visits the bank on a particular date, and signs again, the bank has to check if the signature matches the signature on the records. This task is signature verification. So, the architecture proposed in a Siamese network, as the name says it comes from the Siamese Twins, is to have the same CNN architecture replicated twice, where the signature currently given passes through the same CNN, the signature on the records passes through the same CNN, and you get 2 different representations f_1 and f_2 .

And there has to be a distance measure that checks how close these 2 representations are, which is then passed on to a binary classifier to give the decision of a match or not a match. So, an important takeaway from this approach is that the representations have to be learned in such a way to respect the distance measure that is used to compare these 2 features. So, we will see now that this setting has been improved in many ways over the last.