**Karthik Sajjan**

**Slide 1**

**Slide 2**

Deep learning, a class of machine learning inspired by the structure of human brain, provides solutions using Artificial Neural Networks at the cost of much higher volume of data required to train our machine. The transition from the classic CPUs to the GPUs and large labelled datasets being publicly available allowed significant acceleration in the training of the deep models.

**Slide 3**

Advancements in Deep learning has witnessed the solutions for a variety of computer vision problems, such as object detection, motion tracking, action recognition, human pose estimation, and semantic segmentation.

**Slide 4**

In this context, I wish to draw your attention to the most important type of deep learning model with respect to their applicability in visual understanding known as the Convolutional Neural Networks. We shall also name several other types of deep learning models with respect to their applicability in visual understanding and discuss the strengths and limitations of each of those deep learning models. This will facilitate a compare study with the CNN.

The navigating humanoid employs a facial recognition system that matches a human face from the live video feed through the vision systems against a database of faces used to authenticate users or to recognize them to render various services. We shall discuss the implementation of a face detection and recognition system using for a humanoid using both CNN and Principal Component Analysis.

**Krishna Paanchajanya**

**Slide 5**

Convolutional Neural Networks sometimes referred to as CNN or ConvNet is an artificial neural network model quite popular in analysing visual imagery. A CNN comprises three main types of neural layers, namely, (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers. We shall discuss the role discussed by each and every layer while going through the steps involved in training a CNN deep model with human faces or any other objects.

**Slide 6**

So, coming to the steps involved in training a CNN

Step 1: There are several color spaces in which images exist like Grayscale, RGB, HSV, etc. So, initially, we need to split an image into a maximum number of channels possible.

Step 2: A kernel matrix usually with lesser dimensions than the input image performs dot operation on the image’s pixel matrix sliding by a stride length. As you can see in the figure the kernel colored in orange performs 3x3 matrix dot operations on the 5x5 pixel matrix moving by a stride length 1. This operation is also known as a convolution operation. The layers in which such operations take place are known as convolutional layers. The kernels also known as filters used here are essential in pattern detection in the input image. These patterns can be edges, corners or even more complex objects like human faces, animals, objects, etc. Thus, in the convolutional layers, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. Even the convolution operation can be performed in two ways viz., same padding and valid padding.

**Slide 7**

Step 3: Pooling layers wherein the pooling operations take place are like the in charge of reducing the spatial dimensions of the input image data for the next convolutional layer. The pooling layer does not affect the depth dimension of the volume. The loss in data here is beneficial for the network because the decrease in size leads to less computational overhead for the upcoming layers of the network Average pooling and max pooling are the most commonly used strategies. Also there several other variations of the pooling layer in the literature, like stochastic pooling, spatial pyramid pooling, and def-pooling.

**Step 4:** Following several convolutional and pooling layers, the high-level reasoning in the neural network is performed via fully connected layers. Neurons in a fully connected layer have full connections to all activation in the previous layer, as their name implies. Their activation can hence be computed with a matrix multiplication followed by a bias offset. Fully connected layers eventually convert the 2D feature maps into a 1D feature vector. The derived vector either could be fed forward into a certain number of categories for classification or could be considered as a feature vector for further processing.

Every layer of a CNN thus transforms the input volume to an output volume of neuron activation, eventually leading to the final fully connected layers, resulting in a mapping of the input data to a 1D feature vector. CNNs have been extremely successful in computer vision applications, such as face recognition, object detection, powering vision in robotics, and self-driving cars.

The CNNs achieve significant performance rates in a variety of visual understanding tasks. However, they have their own merits and demerits. CNNs automatically learn features based on the given dataset. CNNs are invariant to transformations, which is a great asset for certain computer vision applications. On the other hand, they heavily rely on the existence of labelled data, in contrast to the other types of deep learning models such as Deep Boltzmann Neural networks which can work in an unsupervised fashion.

**Slide 8**

Face recognition is one of the trending computer vision applications with great commercial interest as well. A feature extractor in a face recognition system extracts features from an aligned face to obtain a low-dimensional representation, based on which an appropriate classifier makes predictions. CNNs brought about a change in the face recognition field owing to their feature learning and transformation invariance properties.

Google’s FaceNet and Facebook’s DeepFace are both based on CNNs. Furthermore, CNNs constitute the core of OpenFace, an open-source face recognition tool, which is of comparable accuracy, and is suitable for mobile computing, because of its smaller size and fast execution time.

**Karthik Sajjan**

**Slide 9**

Principal Component Analysis is a standard technique that is used in statistical pattern recognition and signal processing for data reduction and extraction features.

**Slide 10**

We shall now have a look at the steps to train a machine using PCA for face recognition.

Step 1: Convert the input images to grayscale

Step 2: We implement Histogram equalization method to improve lighting in an image. The areas with lower local contrast can gain a higher contrast in this way.

Consider L as the maximum Gray scale in the pixel matrix of an image, histogram from digital image with Gray scale span 0 to L-1 is a discrete function ℎ(𝑟𝑘) = 𝑛𝑘 where rk is kth Gray scale value, and nk is the number of pixels in image that have rk’s gray scale value.

So, given an image with M x N pixels and L denotes the maximum gray scale value, histogram equalization transformation T can be represented by the function

With the histogram equalization transformation, the lighting in the image can be corrected effectively.

**Slide 11**

Step 3: We flatten all the images available for the training and mention them column wise in a matrix referred to as the face vector space. Flatten in the sense, the pixel matrix of an image unrolls all its values into a vector.

Step 4 and 5: Consider the training set consisting M images each of N x N pixels as Γ = [Γ1, Γ2,… ΓM]. The face vector space is a N2 X M matrix now as each column is a flattened pixel matrix i.e., an image

The mean face is computed as i.e., the average of individual face vectors.

Next, normalizing the face vectors takes place after the face vector space free from common features among all the images which is in turn computed by A = [Φ1, … ΦM] where Φi = Γi – Ψ

The covariance matrix given by C = A.AT leads to a matrix with huge dimensions N2 x N2 wherein computing eigen vectors otherwise known as eigenfaces becomes difficult. Thus, we consider a lower dimensionality subspace here i.e., we calculate C = AT.A instead resulting in a M x M dimension matrix which is smaller and easier to compute compared to the prior covariance matrix.

The set of eigen vectors for C is given as V= [V1, V2, V3, …VM]

**Deepak Chowdary**

**Slide 12**

Step 6 and 7: We pick some best K eigen vectors from the set V, convert them back to the original dimensionality by multiplying them with A, the face vector space with no common features. Thus, the eigen space in the original dimensionality can be considered as some U = [U1, U2, U3, …Uk].

Step 8: Now, all the face vectors from the training set are represented as a linear combination of the K eigenfaces plus the mean face Ψ. Remember, we removed Ψ to eliminate common features? We should put it back now for valid computations. The scalars used in these linear combination form a weight matrix of that particular image from the training set picked.

**Slide 13**

Now that all of the faces in the training set are trained, we shall now see how to recognize an unknown face by the PCA.

* Convert the Input image to a gray scale image
* Improve the lighting using Histogram Equalization
* Convert the input image to a face vector
* Normalize the face vector
* Project the normalized face vector onto the eigen space
* Compute the weight vector of the input image
* Euclidian distance is one of the methods that can be used to match a new face image to the existing face image in the database. Smaller the Euclidian distance, more is the image similar to the one available in the database.
* Calculate the Euclidian distance between this weight vector and the weight vectors of each of the faces already trained.
* The eigen face with which the Euclidian distance is minimum is identified as the person whose face is provided.

**Slide 14**

We have discussed above two methods of facial detection and recognition viz., using Convolutional Neural Networks and Principal Component Analysis. The live video feed perceived by the frontal cameras of the humanoid shall be first converted into grayscale image frames, to reduce computation in the image processing. The CNN deep learning model then convolves the image data across various layers, reduces spatial dimensions using any pooling technique and then recognises the particular person based on the already trained human faces. In the case of PCA, the gray scale image will be initially corrected using histogram equalisation method. The eigenface of the gray scale image and subsequently its weight matrix shall be computed which are further used to calculate the Euclidean distance with respect to the weight matrix of every eigenface in the database to recognize the face. The accuracy of the PCA face recognition is about 93% as per the literature reviewed. Using CNN as classifiers, the face recognition systems reach the accuracy of about 95%. In comparison with this result, the PCA system which uses simple Eulidean distance as classifier has comparable performance with the CNN. However, it is to be noted that CNN is a widely used method in face recognition systems due to its higher accuracy and reliability.

**Slide 15**

Thank you

**Slide 16**

**Slide 17**