Face Recognition

Vincent WHANNOU DE DRAVO, Sidoine AGUÉ, Landry BOSSOU, Joaïda AKOTENOU

La Fondation Vallet & Benin Excellence & UNDP

École d'Été en Intelligence Artificielle (EEIA) - **August 12, 2024**

Presentation Outline

This course was given as part of the fourth edition of École d'Été en Intelligence Artificielle, i.e., the Summer School in Artificial Intelligence initiated by *La Fondation Vallet*, *Benin Excellence*, and *UNDP*, commonly called EEIA.

This presentation is an essay of answers on biometric face recognition technology, their origin, what they are, and how to implement them using machine learning.

Questions about face recognition systems in this course?

- What is face recognition and its importance?
- How does it work?
- Which kind of data are needed?
- How can we build relevant features?
- How can one train a simple face ID model?
- How is a face ID deep learning model built?

Facial Biometry: what is it?

Face Recognition is a biometric technology used to identify or verify a person's identity by analyzing their **facial features**. This process involves capturing one or several images, a 3D representation, or other media of a person's face and then comparing it with stored facial data to find a match.

In which fields can we find it!

Face recognition has become a crucial tool in various applications, including:

- Security systems (e.g., unlocking devices, surveillance)
- Identity verification (e.g., airport security, banking)
- Social media (e.g., tagging friends in photos)
- Law enforcement (e.g., identifying suspects)

Key Milestones and Breakthroughs

- 1960s - Early Concepts: Woodrow W. Bledsoe (1960s) [3]: Considered one of the pioneers, Bledsoe developed some of the earliest facial recognition algorithms. These early systems required humans to manually locate features such as eyes, ears, noses, and mouths on photographs.
- 1970s - Automated Face Recognition: Goldstein, Harmon, and Lesk (1973) [9]: They created a system that used 21 specific subjective markers (e.g., hair color, lip thickness) to recognize faces. This was one of the first attempts to automate face recognition using a computer.
- 1980s - Eigenfaces and Principal Component Analysis (PCA): Lawrence Sirovich and Michael Kirby (1987) [13]: Introduced the concept of using principal component analysis (PCA) to efficiently represent face images. This method reduced the complexity of recognizing faces by transforming the original images into a set of

Key Milestones and Breakthroughs (Cont'n)

- Matthew Turk and Alex Pentland (1991) [16]: Expanded on the work of Sirovich and Kirby by developing a real-time face recognition system using eigenfaces. Their work demonstrated the practical application of PCA in face recognition, leading to significant advancements in the field.
- 1990s - Appearance-Based Methods: Fisherfaces and Linear Discriminant Analysis (LDA) [2]: Techniques such as LDA were developed to enhance the performance of face recognition systems by focusing on the most discriminative features that separate different classes (faces).
- 2000s - The work of Viola and Jones is indeed in the field of face detection, which is a crucial component of face recognition systems [17] - 3D Face Recognition and LBP: Researchers began exploring three-dimensional face recognition to overcome the limitations of 2D methods, such as changes in lighting and pose [4].

Key Milestones and Breakthroughs (Cont'n)

- Local Binary Patterns (LBP) [1]: LBP emerged as a robust texture descriptor used for face recognition.
- 2010s - Deep Learning Revolution: Deep Convolutional Neural Networks (CNNs): The advent of deep learning and convolutional neural networks (CNNs) revolutionized face recognition. Models such as DeepFace (Facebook, 2014) [15] and FaceNet (Google, 2015) [12] achieved unprecedented accuracy by learning hierarchical feature representations directly from raw images. FaceNet (2015): Developed by Google, FaceNet introduced a unified embedding for face recognition, clustering, and verification, setting new standards for accuracy and performance.

Key Milestones and Breakthroughs (Cont'n)

- 2020s - Real-Time and Ethical Considerations: Real-Time Face Recognition: Advances in hardware and algorithms have enabled real-time face recognition on mobile devices and embedded systems. Ethical and Privacy Concerns: With the widespread adoption of face recognition technology, issues related to privacy, bias, and ethical use have become critical topics of discussion.

Face recognition technology generally consists of three phases:

- Preprocessing
- Feature extraction
- Matching

Preprocessing step involves understanding the initial steps of face recognition, including face detection and alignment.

1. Face detection: This is the process of locating and identifying the presence of a face in an image.
2. Alignment: This step adjusts the orientation and size of the face to match a standard template.

Face detection

Face detection is often tackled with one of these three techniques.

1. Haar Cascades: Developed by Viola and Jones [17], it uses a series of simple classifiers to detect faces quickly.
2. Histogram of Oriented Gradients (HOG): Utilizes gradient orientation histograms [5] to detect objects, including faces.
3. Deep Learning: Convolutional Neural Networks (CNNs) can be trained to detect faces with high accuracy.

Face alignment

Face alignment uses mostly the following techniques.

1. Geometric Transformations: Rotates and scales the face to align key facial features (eyes, nose, mouth) to predefined positions.
2. Facial Landmark Detection: Detects key points on the face to aid in precise alignment [19].

Feature extraction

Feature extraction involves learning how distinctive facial features are extracted using various algorithms. Various techniques are used, among them:

1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)
3. Deep Learning Models

Feature extraction: PCA

1. PCA reduces the dimensionality of facial images while preserving the most important information.
2. Example: Used in the Eigenfaces []method to create a set of basis faces.

Feature extraction: LDA

1. LDA finds the linear combinations of features that best separate different classes (e.g., different individuals).
2. Example: Enhances the discriminative power of the features.

Feature extraction: Deep learning models

1. Deep learning models such as Convolutional Neural Networks (CNNs) learn hierarchical feature representations directly from raw images.
2. Example: Models like DeepFace and FaceNet have set new standards in face recognition accuracy.

Matching

The final phase involves matching the extracted features against a database of known faces to find a match or verify an identity.

1. Similarity Measures: Techniques such as Euclidean distance or cosine similarity are used to compare feature vectors.
2. Classification Algorithms: Models such as Support Vector Machines (SVMs) or neural networks are used to classify the input face.

What is Transfer Learning?

Transfer Learning involves applying knowledge gained from solving one problem to a different but related problem.

1. Example: Using a model trained to recognize animals to identify different types of pets.
2. Key Idea: Instead of starting from scratch, we use the patterns learned from one task to improve performance on another.
3. Further reading: [10]

Why Use Transfer Learning in Face Recognition?

Transfer Learning is particularly useful[18] in face recognition because:

1. **Limited Data:** Often, there isn't enough labeled data to train a deep neural network from scratch.
2. **Efficiency:** Pre-trained models can significantly reduce the time and resources required for training.
3. **Improved Accuracy:** Leveraging pre-trained models can enhance the performance of face recognition systems.

Methods for Transfer Learning

Transfer learning methods[6, 11] are basically the following:

1. **Training a Model to Reuse It:**
   - Train on a related task with abundant data and reuse the model for a new task.
   - Example: Train on a large face dataset and fine-tune for a specific application like recognizing employees.
2. **Using a Pre-Trained Model (Fine-Tuning):**
   - Use models like VGGFace or ResNet that are pre-trained on large datasets.
   - Fine-tune the model on your specific dataset by adjusting the latter layers.
3. **Feature Extraction:**
   - The early layers of a deep neural network capture generic features (e.g., edges, textures).
   - The latter layers can be fine-tuned for the specific features relevant to face recognition.

How to do it: Example (Cont'd)

- In transfer learning, we try to transfer as much knowledge as possible from the previous task the model was trained on to the new task at hand. This knowledge can be in various forms depending on the problem and the data. For example, it could be how models are composed, which allows us to more easily identify novel objects.

What methods are used: Details

Basically, 3 types (See above):

- 1. Training a Model to Reuse it
  Suppose you need to address task A but lack sufficient data to train a deep neural network. A practical approach is to identify a related task B that has ample data available. You can train the deep neural network on task B, and then use this trained model as a foundation for solving task A. Depending on the specifics of your problem, you may either utilize the entire model or focus on reusing only certain layers.
  If both tasks share similar input types, reusing the model to make predictions on your new input might be feasible. Alternatively, you could modify and retrain specific layers tailored to the new task, particularly the output layer.

What methods are used: Details(Cont'd)

- 2. Using a Pre-Trained Model (fine-tuning)
  The second method involves using a pre-trained model. There are many such models available, so it's important to do some research. The decision on how many layers to reuse and how many to retrain will vary based on the specific problem you're tackling.
  For instance, Keras offers a variety of pre-trained models that can be utilized for transfer learning, prediction, feature extraction, and fine-tuning. You can explore these models, along with tutorials on how to use them, online. Additionally, many research institutions publish trained models for public use.
  This approach to transfer learning is widely employed in deep learning.

What methods are used: Details(Cont'd)

- 3. Feature Extraction
  Another method is to use deep learning to automatically identify the most important features, a process known as representation learning. This often outperforms manually crafted features. While feature engineering and domain knowledge are still important, deep learning allows neural networks to determine which features are crucial and which are not.

What are Hyperparameters?

- **Hyperparameters** are settings or configurations used to control the training process of a model[8]. They are the variables which determine the network structure and the variables which determine how the network is trained (e.g., learning rate). Hyperparameters are set before training (before optimizing the weights and bias).
  - **Network Structure Hyperparameters:** Define the architecture of the model (e.g., number of layers, type of activation functions).
  - **Training Algorithm Hyperparameters:** Control the learning process (e.g., learning rate, batch size, number of epochs).

Network Structure Hyperparameters

More on network structure Hyperparameters in [14]

- Hidden layers: These are the layers between the input layer and the output layer.
  "Very simple. Just keep adding layers until the test error does not improve anymore."
  Many hidden units within a layer with regularization techniques can increase accuracy. A smaller number of units may cause underfitting.
- Dropout: Dropout is a regularization technique to avoid overfitting (increasing the validation accuracy) thus increasing the generalizing power. Generally, use a small dropout value of 20%-50% of neurons with 20% providing a good starting point. A probability too low has minimal effect and a value too high results in under-learning by the network. Use a larger network. You are likely to get better performance when dropout is used on a larger network, giving the model more of an opportunity to learn independent representations.

Hyperparameters related to Activation Function and Weight

This topic is covered in [7]

- Network Weight Initialization: Ideally, it may be better to use different weight initialization schemes according to the activation function used on each layer. Mostly uniform distribution is used.
- Activation functions: These are used to introduce nonlinearity to models, which allows deep learning models to learn nonlinear prediction boundaries. Generally, the rectifier activation function is the most popular. Sigmoid is used in the output layer while making binary predictions. Softmax is used in the output layer while making multi-class predictions.

Hyperparameters related to Training Algorithm

- Learning Rate: This defines how quickly a network updates its parameters. A low learning rate slows down the learning process but converges smoothly. A larger learning rate speeds up the learning but may not converge. Usually, a decaying learning rate is preferred.
- Momentum: This helps to know the direction of the next step with the knowledge of the previous steps. It helps to prevent oscillations. A typical choice of momentum is between 0.5 to 0.9.

Hyperparameters related to Training Algorithm (Cont'd)

- Number of Epochs: This is the number of times the whole training data is shown to the network while training. Increase the number of epochs until the validation accuracy starts decreasing, even when training accuracy is increasing (overfitting).
- Batch Size: Mini batch size is the number of subsamples given to the network after which parameter update happens. A good default for batch size might be 32. Also try 32, 64, 128, 256, and so on.

What are the benefits?

We let you guess them!

Thanks

Thank you!

List of References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. "Face Description with Local Binary Patterns: Application to Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 2037–2041.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), pp. 711–720.

[3] Woodrow W. Bledsoe. *The Model Method in Facial Recognition*. Tech. rep. PRI-TR-1974-01. Available at `http://example.com/techreport`. Palo Alto, California, USA: Panoramic Research Inc., Mar. 1974.

List of References

[4] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. "Three-dimensional face recognition". In: *International Journal of Computer Vision* 64.1 (2005), pp. 5–30.

[5] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, pp. 886–893.

[6] Jeff Donahue et al. "Decaf: A deep convolutional activation feature for generic visual recognition". In: *International conference on machine learning*. PMLR. 2014, pp. 647–655.

List of References

[7] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[9] L.D. Harmon and A.B. Lesk. "Automatic recognition of human face profiles". In: *Computer Graphics and Image Processing* 2.1 (Apr. 1973), pp. 18–34.

[10] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

List of References

[11] Ali Sharif Razavian et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 806–813.

[12] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823.

[13] L. Sirovich and M. Kirby. "Low-dimensional procedure for the characterization of human faces". In: *Journal of the Optical Society of America* 4.3 (1987), pp. 519–524.

List of References

[14] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[15] Y. Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1701–1708.

[16] M. Turk and A. Pentland. “Face recognition using eigenfaces”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, pp. 586–591.

List of References

[17] P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2001, pp. 511–518.

[18] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[19] Z. Zhang et al. "Facial landmark detection by deep multi-task learning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2014, pp. 94–108.