https://doi.org/ 10.33472/AFJBS.6.10.2024.4694-4701

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

About Face Recognition System Artificial Intelligence Technology

Gulnora Mirazimova Xasanovna,
Almardanov Mukhriddin Khurram o‘g‘li

Uzbekistan, Tashkent, Amir Temur st. 108, TUIT named after Muhammad al-Khwarizmi

Corresponding author: gmirazimova1974@gmail.com, almardanovmail.com@gmail.com.

Article History

Volume 6,Issue 10, 2024

Received:29 Apr 2024

Accepted : 27 May 2024

doi: 10.33472/AFJBS.6.10.2024.4694-4701

Abstract. In this work, thesis focuses on the research and development of a facial recognition system based on neural networks. The work begins with an overview of existing facial recognition methods, highlighting their relevance and significance. The theoretical review of facial recognition methods encompasses the fundamentals of neural networks and artificial intelligence.

RELEVANCE OF THE TOPIC

Using machine learning and neural networks, artificial intelligence can be trained to recognize and classify images. This has a wide range of applications, including security and surveillance, image scanning and creation, marketing and advertising, augmented reality, and image search. Perhaps the most popular and promising task of neural networks is image recognition technology. They are used either individually or in integrated form in such areas as security and surveillance, image scanning and creation, marketing, and advertising.

Teaching a computer to recognize objects in images is a complex task because the computer sees the world differently than humans. To teach a computer to recognize and understand images, machine learning methods are used. This includes collecting large databases and forming datasets, extracting features and combinations for object identification, and then training machine learning models on this data.

The process of image recognition in neural networks is carried out by dividing the image into small sections and transmitting signals between neurons. Neurons compare the received signals with processed data previously learned by the network. Thus, the neural network can determine how similar the images are and classify objects. Using neural networks and machine learning allows the creation of systems capable of recognizing and classifying images. These systems facilitate human work and increase the accuracy and reliability of various work processes by eliminating the human factor.

In the field of video surveillance, for example, systems analyze images and recognize objects, and then artificial intelligence classifies actions. For image recognition, the neural network must be trained on relevant data. It is also important to have a sufficiently large and high-quality dataset for training the neural network. A larger size and quality of the dataset allow achieving higher recognition accuracy.

ANALYSIS OF NEURAL NETWORKS AND ARTIFICIAL INTELLIGENCE

Research in the field of artificial intelligence-based facial recognition systems using deep learning has high relevance and a wide range of applications. Here are several key aspects highlighting its significance:

Security and providing secure environments: Deep learning-based facial recognition systems provide effective solutions for access control, crime detection, and prevention.

Improvement of user interfaces: Facial recognition can significantly enhance user experience and comfort in various devices and applications. This allows the use of facial biometric authentication to unlock devices, access systems, and authorize payments without the need to remember passwords or use other authentication methods.

Marketing and personalization: Facial recognition systems can be applied to collect and analyze consumer behavior data, which helps improve marketing strategies and provide personalized content and services. This enables companies to better understand their customers, offer relevant recommendations, and increase the effectiveness of advertising campaigns.

Research in the field of neural network-based facial recognition systems is driven by their wide range of applications and potential to enhance security technologies, user convenience, marketing strategies, and the development of autonomous systems. It has been noted that this area is active and rapidly evolving, requiring further research and innovation.

The fundamentals of neural networks and artificial intelligence are based on modeling the workings of the human brain and the development of computer systems capable of learning and making decisions based on data. Neural networks are mathematical models that mimic the functioning of neurons in the brain. They consist of a multitude of interconnected artificial neurons that transmit and process information.

Based on neural networks and other machine learning methods, AI systems can recognize images, process natural language, perform automated tasks, and make decisions in complex situations.

The basics of neural networks and artificial intelligence are fundamental concepts and methods that underlie the development and application of modern technologies and systems. They find wide application in various fields such as computer vision, natural language processing, process automation, and decision-making. Understanding these fundamentals enables the creation of more efficient and intelligent systems capable of analyzing and processing data with high accuracy and speed.

Neural networks are used to solve complex tasks that require analytical calculations similar to those performed by the human brain. The most common applications of neural networks include:

- Classification: Distribution of data by parameters. For example, given a set of people, deciding whom to grant credit to and whom not.
- Prediction: The ability to predict the next step. For example, the rise or fall of stocks based on the situation in the stock market.
- Recognition: Currently, the most widespread application of neural networks. It is used in Google when you search for photos or in phone cameras when it detects the position of your face and highlights it, and much more.

Basics of Neural Networks and Artificial Intelligence

Main components of a neural network:

- Input data: This is the information fed into the neural network for processing. It can be represented as numbers, images, text, etc.
- Network architecture: The architecture of a neural network determines how neurons and layers are interconnected. It can be simple, such as a single-layer network, or complex, consisting of multiple layers and connections.
- Training: Neural networks are trained by adjusting weights and biases based on training data. The training process involves computing the network's error and using optimization algorithms to minimize this error.
- Preprocessing of facial images (Fig. 1, 2):
- Resize: Facial images may have different sizes, and to ensure uniformity in processing, they should be resized to a consistent size.
- Cropping: Sometimes it is necessary to isolate only the face in the image and remove background elements. This can be done by cropping the image to leave only the area containing the face. Facial detection methods such as convolutional neural network-based algorithms or Haar cascades can be used for this purpose.

- Color transformation: Depending on the source and conditions of the image, they may be in different color spaces. Color transformation can be useful for standardizing the color characteristics of images and removing distortions.

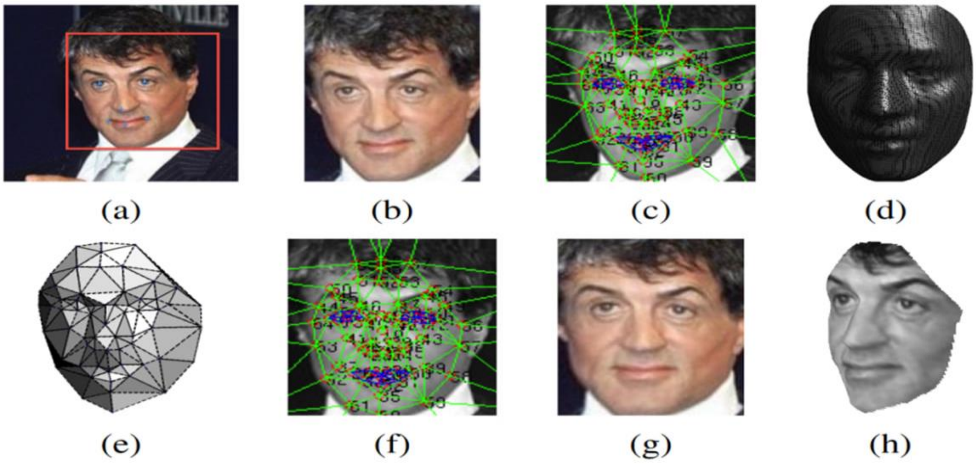
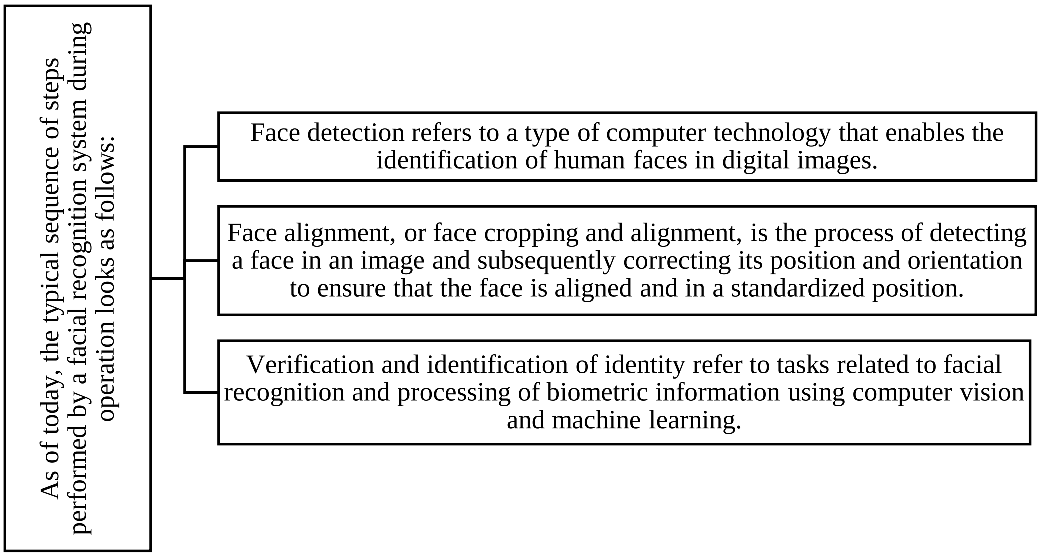


FIGURE 1. Process of three-dimensional face alignment using key points


As of today, the typical sequence of steps performed by a facial recognition system during operation looks as follows:

Face detection refers to a type of computer technology that enables the identification of human faces in digital images.

Face alignment, or face cropping and alignment, is the process of detecting a face in an image and subsequently correcting its position and orientation to ensure that the face is aligned and in a standardized position.

Verification and identification of identity refer to tasks related to facial recognition and processing of biometric information using computer vision and machine learning.

FIGURE 2. Scheme of the process of three-dimensional face alignment using key points

To develop a facial recognition system, it is necessary to choose a suitable neural network architecture. Various options are considered, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), or their combinations. When choosing the architecture, requirements for accuracy, processing speed, and system resource consumption are taken into account. Preprocessing of images is required before feeding them into the neural network.

Technologies and Tools for Algorithm Development and Implementation

In the field of facial recognition, various technologies and tools are used for the development and implementation of algorithms. Here are some of them:

Python is one of the most popular programming languages in the field of artificial intelligence and machine learning. It offers a wide range of libraries and frameworks that simplify the development and implementation of facial recognition algorithms, such as TensorFlow, Keras, PyTorch, and OpenCV.

Here is an example of simple Python code that uses the OpenCV library for facial recognition on an image:

```
import cv2
# Loading a face classifier
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
# Loading an image
image = cv2.imread('image.jpg')
# Converting the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Detecting faces in the image
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
# Drawing rectangles around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 3)
# Displaying the result
cv2.imshow('Face Detection', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In this example, we are using the haarcascade_frontalface_default.xml face classifier, which is part of the OpenCV library. We load the image, convert it to grayscale, and then apply the detectMultiScale method to detect faces in the image. Then we draw rectangles around the detected faces and display the result. This is just one example of how Python and OpenCV can be used for facial recognition. Python also supports other libraries such as TensorFlow, Keras, and PyTorch, which offer more advanced capabilities for training and deploying neural networks in facial recognition tasks.

TensorFlow is an open-source deep learning platform developed by Google. It provides extensive capabilities for creating and training neural networks, including convolutional neural networks (CNNs). TensorFlow enables efficient processing of large datasets and development of complex facial recognition models.

Here is an example Python code that uses the TensorFlow library to create and train a convolutional neural network for facial recognition:

```
import tensorflow as tf
from tensorflow.keras import layers
# Creating a convolutional neural network (CNN) model
model = tf.keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])
# Compiling the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
# Loading the training data
train_data = tf.keras.preprocessing.image_dataset_from_directory(
    'dataset/train',
    image_size=(64, 64),
    batch_size=32,
    label_mode='binary'
)
# Training the model
model.fit(train_data, epochs=10)
# Loading the testing data
test_data = tf.keras.preprocessing.image_dataset_from_directory(
    'dataset/test',
    image_size=(64, 64),
    batch_size=32,
    label_mode='binary'
)
```

```
# Evaluating the model on the testing data
loss, accuracy = model.evaluate(test_data)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')
```

In this example, we create a convolutional neural network (CNN) model using the TensorFlow library. We define convolutional layers, pooling layers, and fully connected layers that make up the model. Then, we compile the model with an optimizer, a loss function, and metrics. We load the training and testing data using `image_dataset_from_directory`, specifying the image size and batch size. Next, we train the model on the training data using the `fit` method. After training, we load the testing data and evaluate the model on it using the `evaluate` method. The results, such as loss and accuracy, are displayed on the screen.

Keras is a high-level deep learning framework built on top of TensorFlow. It provides a simple and intuitive interface for creating and training neural networks. Keras offers a wide range of functionalities and is widely used in developing face recognition models.

This Python code example utilizes the Keras library with TensorFlow as the backend to create and train a face recognition model:

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
# Creating a neural network model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compiling the model
model.compile(optimizer='adam',
      loss='binary_crossentropy',
      metrics=['accuracy'])
```

In this code, a sequential model of a neural network is created, consisting of several layers including convolutional layers (Conv2D), pooling layers (MaxPooling2D), flattening layer (Flatten), and fully connected layers (Dense). After training, we load the testing data and evaluate the model.

RESULT. Attendance Tracking of Students Recognized by Their Facial Features

Below is the program primarily acting as a simple attendance system where students' attendance is marked based on their presence in front of the webcam, recognized by their facial features:

```
import face_recognition
import cv2
import numpy as np
import csv
import os
from datetime import datetime
video_capture = cv2.VideoCapture(0)
jobs_image = face_recognition.load_image_file("photos/Gates.jpg")
jobs_encoding = face_recognition.face_encodings(jobs_image)[0]
ratan_tata_image = face_recognition.load_image_file("photos/DiCaprio.jpg")
ratan_tata_encoding = face_recognition.face_encodings(ratan_tata_image)[0]
sadmona_image = face_recognition.load_image_file("photos/Musk.jpg")
sadmona_encoding = face_recognition.face_encodings(sadmona_image)[0]
tesla_image = face_recognition.load_image_file("photos/Tesla.jpg")
tesla_encoding = face_recognition.face_encodings(tesla_image)[0]
tesla2_image = face_recognition.load_image_file("photos/Bezos.jpg")
```

```
 tesla_encoding2 = face_recognition.face_encodings(tesla2_image)[0]
 tesla3_image = face_recognition.load_image_file("photos/AmirTemur.jpg")
 tesla_encoding3 = face_recognition.face_encodings(tesla3_image)[0]
 known_face_encoding = [
 jobs_encoding,
 ratan_tata_encoding,
 sadmona_encoding,
 tesla_encoding,
 tesla_encoding2,
 tesla_encoding3
 ]
 known_faces_names = [
 "Gates",
 "DiCaprio",
 "Musk",
 "Tesla",
 "Jeff Bezos",
 "Amir Temur"
 ]
 students = known_faces_names.copy()
 face_locations = []
 face_encodings = []
 face_names = []
 s=True
 now = datetime.now()
 current_date = now.strftime("%Y-%m-%d")
 f = open(current_date+'.csv','w+',newline = '')
 lnwriter = csv.writer(f)
 while True:
_,frame = video_capture.read()
small_frame cv2.resize(frame,(0,0),fx=0.25,fy=0.25)
rgb_small_frame = small_frame[:,:,::-1]
if s:
    face_locations face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame,face_locations)
    face_names = []
    for face_encoding in face_encodings:
        matches face_recognition.compare_faces(known_face_encoding,face_encoding)
        name=""
        face_distance = face_recognition.face_distance(known_face_encoding,face_encoding)
        best_match_index = np.argmin(face_distance)
        if matches[best_match_index]:
            name = known_faces_names[best_match_index]

        face_names.append(name)
        if name in known_faces_names:
            font = cv2.FONT_HERSHEY_SIMPLEX
            bottomLeftCornerOfText = (10,100)
            fontScale              = 1.5
            fontColor              = (255,0,0)
            thickness              = 3
            lineType               = 2

            cv2.putText(frame,name+' Present',
                bottomLeftCornerOfText,
                font,
                fontScale,
                fontColor,
                thickness,
                    lineType)

                if name in students:
                    students.remove(name)
                    print(students)
                    current_time = now.strftime("%H-%M-%S")
                    lnwriter.writerow([name,current_time])
    cv2.imshow("attendence system",frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.destroyAllWindows()
f.close()
```

CONCLUSION

The use of neural networks in facial recognition systems provides more accurate and reliable recognition, overcoming limitations associated with human factors. Various facial recognition methods represent different approaches and are based on analyzing facial geometric features, using face templates, or deep learning with neural networks. Each of these methods has its advantages and limitations, and the choice of a specific method depends on the requirements and tasks.

Neural network-based facial recognition systems have a wide range of applications in security and surveillance, image scanning and creation, marketing and advertising, augmented reality, and image search. The relevance of this research is due to the potential of neural networks to improve security technologies, user convenience, marketing strategies, medical diagnostics, and the development of autonomous systems. However, this field is active and rapidly evolving, requiring further research and innovation to achieve even higher results and overcome current limitations.

Thus, neural network-based facial recognition systems represent a promising research direction. The study of the fundamentals of neural networks and artificial intelligence has covered the basic principles of neural networks, including perceptrons, activation functions, forward and backward error propagation. Artificial intelligence includes machine learning and deep learning, which play an important role in developing facial recognition systems.

Facial recognition methods based on neural networks offer significant advantages in accuracy and the ability to learn from large volumes of data. However, achieving optimal results requires selecting the right neural network architecture, optimizing hyperparameters, and training on diverse datasets.

An overview of the main algorithms and approaches in facial recognition has identified various methods, such as principal component analysis (PCA), linear discriminant analysis (LDA), methods based on convolutional neural networks (CNN), and recurrent neural networks (RNN). Each method has its advantages and limitations, and the choice of approach depends on specific requirements and application contexts.

The study also explored the technologies and tools used in facial recognition, including popular libraries and frameworks for developing neural networks such as TensorFlow and PyTorch, as well as existing facial databases used for training and testing recognition systems.

Based on the conducted review, it can be concluded that neural network-based facial recognition methods represent an efficient and promising approach.

REFERENCES

1. Z. Zhang and Z. Zhang, A Review on Face Detection and Recognition Techniques. (Artificial Intelligence Review, 47(3), 2016), pp. 346-364.
1. Y. Taigman, M. Yang, M. Ranzato and L. Wolf, DeepFace: Closing the gap to human-level performance in face verification. (In Proceedings of the IEEE conference on computer vision and pattern recognition, 2014), pp. 1701-1708.
2. F. Schroff, D. Kalenichenko and J. Philbin, FaceNet: A unified embedding for face recognition and clustering. (In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015), pp. 815-823.
3. O. M. Parkhi, A. Vedaldi, and A. Zisserman, Deep face recognition. (In Proceedings of the British Machine Vision Conference, 2015), pp. 41.1-41.12.
4. J. Deng, J. Guo and S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition. (In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019), pp. 4690-4699.

5. W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj and L. Song, SphereFace: Deep hypersphere embedding for face recognition. (In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017), pp. 212-220.
6. J. Deng, J. Guo and S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition. (In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019), pp. 4690-4699.
7. A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. (In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015), pp. 3128-3137.