International Journal of Computational Intelligence Systems (2022) 15:49
https://doi.org/10.1007/s44196-022-00108-2

RESEARCH ARTICLE

An Eye State Recognition System Using Transfer Learning: AlexNet-Based Deep Convolutional Neural Network

Ismail Kayadibi[1] · Gür Emre Güraksın[2] · Uçman Ergün[1] · Nurgül Özmen Süzme[1]

Received: 18 February 2022 / Accepted: 8 July 2022

Abstract

For eye state recognition (closed or open), a mechanism based on deep convolutional neural network (DCNN) using the Zhejiang University (ZJU) and Closed Eyes in the Wild (CEW) dataset, has been proposed in this paper. In instances where blinking is consequential, eye state recognition plays a critical part for the development of human–machine interaction (HMI) solutions. To accomplish this objective, pre-trained CNN architectures on ImageNet were first trained on the both dataset, which included both open and closed-eye states, and then they were tested, and their performance was quantified. The AlexNet design has proven to be more successful owing to these assessments. The ZJU and CEW datasets were leveraged to train the DCNN architecture, which was constructed employing AlexNet modifications for performance enhancement. On the both datasets, the suggested DCNN architecture was tested for performance. The achieved DCNN design was found to have 97.32% accuracy, 95.37% sensitivity, 97.97% specificity, 93.99% precision, 94.67% $F1$ score, and 99.37% AUC values in the ZJU dataset, while it was found to have 97.93% accuracy, 98.74% sensitivity, 97.15% specificity, 97.11% precision, 97.92% $F1$ score, and 99.69% AUC values in the CEW dataset. Accordingly, when compared to CNN architectures, it scored the maximum performance. At the same time, the DCNN architecture proposed on the ZJU and CEW datasets has been confirmed to be an acceptable and productive solution for eye state recognition depending on the outcomes compared to the studies in the literature. This method may contribute to the development of HMI systems by adding to the literature on eye state recognition.

Keywords Eye state recognition · Human–machine interaction · Deep learning · Deep convolutional neural network · Transfer learning

Abbreviations

ZJU	Zhejiang University
DCNN	Deep convolution neural network
CEW	Closed eyes in the wild
HMI	Human–machine interaction
CVS	Computer vision syndromes
ECG	Electrocardiogram
EEG	Electroencephalogram
EOG	Electrooculogram
SVM	Support vector machine
HOG	Histogram oriented gradient
ANN	Artificial neural network
CNN	Convolutional neural network
LKT	Lucas–Kanade–Tomasi
LTP	Local ternary patterns
MultiHPOG	Multi-scale histograms of principal oriented gradients
LBP	Local binary pattern
Multi-TPLBP	Multi-three-patch local binary pattern histogram
WBCNNTL	Weight binarization convolution neural network and transfer learning
DCNNE	Dual convolution neural network ensemble
DFNN	Deep-fusion neural network
ReLU	Rectified linear unit
FC	Fully connected layers
TP	True positive
TN	True negative
FP	False positive
FN	False negative

✉ Gür Emre Güraksın
emreguraksin@aku.edu.tr

[1] Department of Biomedical Engineering, Afyon Kocatepe University, Afyonkarahisar, Turkey

[2] Department of Computer Engineering, Afyon Kocatepe University, Afyonkarahisar, Turkey

Published online: 22 July 2022

ROC	Receiver-operating characteristic
AUC	Area under curve

1 Introduction

One of the facial traits used to help decide whether an eye is open or closed is the state of the eye. Furthermore, it is a fundamental criterion that accurately depicts a person's physiological state. Even though eye state can be described in a variety of ways, it can generally be divided into two categories: open and closed. It has great potential in many fields such as drowsiness recognition [1], facial expression recognition [2, 3], liveness detection [4], and eye fatigue estimation [5]. In addition, eye state is a constructive instrument for establishing HMIs and is commonly utilized in computer vision systems [6]. Many people are susceptible to ocular symptoms such as dry eye triggered by computer use as a byproduct of recent technological advancements and the fact that computers have become a part of our daily life [7]. Computer vision syndromes (CVS) are a group of symptoms produced by people's inability to adjust their eye state (e.g., blinking) while also staring at digital screens for long periods of time. Eye state recognition plays a crucial role in recognizing a person's blinking state in front of the screen for this purpose. On a digital screen, having a minimal number of blinks has both beneficial and bad implications. While the advantageous consequences of blinking correlate to attention diversion and perception on screens, the detrimental repercussions are associated with human health and are concerning since they amount to an increase in the number of people affected by CVS [8]. Eye state recognition has grown extremely influential in the field of computer vision. It makes a significant contribution to the advancement of human–computer interaction [9] technology by allowing for accurate eye state and blink recognition. In addition, there has been a surge in interest in eye state studies since the recognition of eye state boosts awareness in many domains.

The state of one's eyes can also be utilized to determine driver fatigue. Driver fatigue is detected using a variety of approaches, including the monitoring of controlled equipment, physiological indicators, and actions. Because of the high reliance on driver skills and road quality, monitoring programmable equipment is a non-invasive method with limited reliability. Screening controllable equipment necessitates the driver attaching signal measuring devices to his body, making it almost impossible to observe these physiological indications. Eye characteristics such as the degree of eye opening and the number of blinks are used in behavioral and computer vision measures to diagnose fatigue [10].

Among the most frequent causes of catastrophic car accidents is the driver drowsiness (insomnia, fatigue, inattention, and so on). Detecting driver drowsiness could be a major component of driverless vehicles in the future. Drowsiness in drivers may be identified using a number of methods, and might be divided into three categories: physiological, vehicle-based, and behavioral [10]. Physiological parameters such as the electrocardiogram (ECG), electroencephalogram (EEG), and electrooculogram (EOG) obtained from sensitive electrodes or electronic devices put on the driver are instances of physiological measurements. Physiological measurements, on the other hand, are not generally employed because they impede the driver. Monitoring the vehicle's operated equipment (steering wheel, lane monitoring, and brake regulations) entails a high reliance on driver abilities and road conditions. This is another non-invasive drowsiness detection method with a low level of accuracy. Since behavioral perspectives emphasize on the person rather than the resource, they are more trustworthy than physiological and tool-based methods. They rely on computer vision systems that assess the driver's movement, facial expression, eye state, and blink status using video-recorded visual cues to identify fatigue. Behavioral techniques have recently gained popularity due to their lack of invasiveness and concentration on the driver [11].

Thanks to recent breakthroughs in fields including face and eye recognition and tracking [12, 13], machine learning [14, 15], feature extraction [16], and deep learning [17], substantial progress has been achieved in eye state recognition. Notwithstanding, it is still evolving on a daily basis because eye state recognition encompasses so many characteristics. Early literature on eye state recognition relied on three scenarios: feature-based [18, 19], motion-based [20], and appearance-based [21]. Geometric features and gray-level patterns are used in feature-based methodologies. The properties of eyelid movement are the focus of movement-based approaches. Tissue aspects of the eye area are addressed in appearance-based techniques. The results of experiments suggest that view-based strategies outperform alternative methods [22]. Nonetheless, environmental considerations have a vital effect in accurately defining the eye condition. Many challenges and external elements, such as lighting, light angle, head posture, and image quality, may generate a considerable impact in the appearance and shape of the eyes, making it difficult to accurately quantify the eye state [12]. Since the real world is noisy and new environments are surprisingly uncontrollable. Machine learning algorithms such as AdaBoost [13] and support vector machine (SVM) [15] have been proposed in previous publications on eye state recognition to enhance the efficiency of recognition systems in unpredictable (uncertain) contexts. However, in addition to these machine learning methods, manual feature extraction methods must be used to retrieve the features. Furthermore, because hand-crafted feature extraction approaches require a lot of computing, the produced systems are not only slow, but they also necessitate a lot of expertise and experience [22].

Artificial intelligence techniques, particularly the deep learning method, have shown superior performance in solving problems in many areas as computing, storage, and human–computer interaction technologies have accelerated in the past few years, and it has been demonstrated that the deep learning method has learning algorithms that can effectively distinguish specific (unique) scenarios [17]. Deep learning is a machine learning technology that incorporates neural networks with multiple layers in their structure to approximate the properties of the human brain's nervous system. After extracting the representative characteristics of the raw data, this method integrates all low-level features and may automatically assemble a high abstract representation. In consequence, in light of current breakthroughs, there is no need to manually extract representative characteristics from raw data. But even so, because eye state recognition is used in a wide range of contexts and hardware, including computers for eyestrain detection, vehicles for driver drowsiness detection, and devices encompassing human–computer interaction for eye blink or facial expression detection, several strategies to specifying eye state recognition have been proposed in the last decade.

Dong et al. [11] adopted Random Forest, Random Ferns, and SVM approaches to categorize the feature sets generated from various feature extraction methods for eye state definition. They claimed that the histogram-oriented gradient (HOG) was less influenced by the noise effect for classification purposes on grounds of these classifications and that their approach had a success rate of up to 93% [11]. To detect blinking in low-resolution eye images, Pauly and Sankar [18] applied multiple features (mean intensity, Fisher faces, and HOG feature) and classifiers such as SVM and artificial neural network (ANN). The features acquired by the HOG outperformed all other approaches in the study when utilized with the SVM classifier, according to the comparative results of the five distinct methods used in the study [18]. Pauly and Sankar [23] proposed a method for eye tracking and blink detection in video frames obtained from webcams. This method includes a haar-cascade classifier for eye tracking and a combination of SVM classifier and HOG features for blink detection. The investigators evaluated the proposed blink detection algorithms on images from two available to public datasets (ZJU and CEW) and found that they were 92.5% accurate on average. For blink detection or eye tracking on smartphone platforms, Han et al. [24] suggested a hybrid strategy integrating two machine learning algorithms (SVM and CNN). They also employed multi-class SVM as an alternative to the proposed hybrid technique and evaluated it by comparing it to the hybrid method. They discovered that the LeNet-5 CNN model outperformed the multi-class SVM approach and other methods in a comparison of the presented methods for characterizing blinks. They also made a point of saying that the linear SVM classifier and the LeNet-5 CNN model with HOG features may be utilized to capture blinks in mobile environments efficiently and reliably [24]. Lee et al. [19] used both the AdaBoost face detector and the Lucas–Kanade–Tomasi (LKT) method to detect the face and eye regions. They introduced a feature-based strategy employing the width and height properties of the eye regions to ascertain whether the eye is open or closed in the SVM classifier after calculating the regions with these methods [19]. For the definition of eye state recognition, Zhao et al. [22] presented a deep integrated neural network based on classification according to actionable information in the eye region. They have attempted several configurations by adjusting the training types in this integrated neural network and alleged that it makes the highest performance, allowing them to boost the ability to categorize in small datasets combining the transfer learning and the data augmentation. Song et al. [12] proposed a feature-based method for detecting eye closeness and extracted the features of eye patches using HOG, Local Ternary Patterns (LTP) and Gabor wavelets methods. In order to boost resilience against image disturbances and scale variations, they suggested the novel feature descriptive multi-scale histograms of principal oriented gradients (MultiHPOG) approach. The SVM classifier was used to classify diverse feature fusion schemes in this investigation, and the feature-based SVM classifier with the combination of MultiHOPG, LTP, and Gabor features produced the desired results. To detect driver drowsiness, Wu et al. [1] employed the local binary pattern (LBP) approach and suggested the feature-based SVM classifier method. They concluded that this method might productively distinguish eye state recognition and driver drowsiness after analyzing the testing findings. Liu et al. [25] used appearance-based detection (LBP, Gabor wavelets and HOG) methods for eye closeness detection to extract major components of the eye. Nearest Neighbor, SVM, and Adaboost algorithms were used to designate the constituents obtained by these approaches. The feature set that incorporates the usage of LBP, HOG, and Gabor wavelets combined was identified by SVM technique as the most effective and satisfactory performance among the numerous feature combination schemes given in the study. Eddine et al. [21] used a plethora of feature sets for feature extraction from the eye region in eye localization and state recognition, with the multi-three-patch local binary pattern histogram (Multi-TPLBP) technique of feature extraction with the radial basis function-based SVM classifier attaining the optimal accuracy. Huang et al. [10] presented a deep learning-based convolutional neural network-based drowsiness detection method. The Weight Binarization Convolution Neural Network and Transfer Learning (WBCNNTL) methods were proposed by Liu et al. [26] for eye state definition, with the intention of helping to improve both the learning time and the accuracy of the system. Wang et al. [27] tried to determine the most robust classifier using different classifiers (Ridge regression, SVM, AdaBoost, Stacked Autoencoders, and CNN) to detect the open and closed-eye state. They applied feature descriptors (projections, LBP, HOG) for feature extraction for closed-eye detection. As a result, they performed these operations on the ZJU dataset and stated that among the classification models the automatic encoder model based on the HOG feature achieved the best performance. Saurav et al. [28] proposed a vision-based system for real-time eye state recognition on an embedded platform. In this proposed system, to overcome the overfitting problem created by deep neural networks in small datasets, a dual convolution neural network ensemble (DCNNE) model is developed by combining two lightweight CNNs based on transfer learning-based fine-tuning. This method was validated on three eye condition datasets (ZJU, CEW, and MRL) and experimental results indicated that the proposed DCNNE method showed remarkable success in CEW and ZJU dataset compared to the state-of-the-art methods in the literature. Liang et al. [29] used the deep-fusion neural network (DFNN) model, which is formed by combining the deep neural network, which extracts the vector features of the eye, and the deep convolutional neural network, which extracts the tissue features, to increase the detection efficiency and accuracy in the detection of eye fatigue in controllers. The proposed method was evaluated in ZJU, CEW and ATCE eye state datasets, and the comparative results showed that DFNN outperformed the early technologies used in eye state recognition.

The evolution of deep learning methods, as well as contemporary achievements in artificial intelligence, has enabled the development of new methods and ideas in image categorization. Because of the superior performance of CNN in image classification, one of the sub-branches of machine learning has displayed a considerable effect in many image-based applications [17]. As a consequence, in research by Liu et al. [26], Huang et al. [10], Saurav et al. [28] and Liang et al. [29], deep learning-based CNN approaches, have gotten started to be preferred in eye state recognition. Unlike other machine learning approaches, CNN's multi-layered structure does not demand meticulous engineering or knowledge because it yields representational learning from its raw data. This is because CNNs have outperformed traditional machine methods in studies such as estimating the activity of potential drug molecules and predicting the effects of DNA mutations in raw data comprising high-dimensional capacity image and feature vectors, notably in image recognition and speech recognition. They have also outperformed traditional machine methods in research findings, such as predicting the activity of potential drug molecules and inferring the effects of mutations in DNA [17]. It receives the raw data given to its input as input and autonomously uncovers representative features required for classification by filtering in a similar way to pixel processing using the representation learning structure it contains in the CNN structure. The features extracted from the raw data are mirrored in the network's output, culminating in a representation of the intended classifications. The representative features of CNN can be more thorough than those produced manually by conventional machine learning approaches due to the automatic gathering of each piece of raw data. On account of that, rather than the hand-crafted feature extraction approaches performed in prior studies, the application of deep learning-based, particularly transfer learning-based methods in eye state detection, has emerged as a promising capability in terms of both speed and utility [10, 26]. Transfer learning is an approach to dealing with modest changes between datasets by applying the knowledge learnt by a neural network from one task to another independent learning assignment [30]. In countless fields, such as medical image analysis, transfer learning is favored when there is inadequate data in the datasets during the learning process [30, 31]. ImageNet [32], a large image database dedicated for use in visual object recognition software research, hosted a competition in 2012, and since then, the outstanding success of CNN methods has substantially extended its application in the field of computer vision. CNN models have been developed on ImageNet with step-by-step advances in recent decades, and also many pre-trained models have been constructed, including AlexNet [33], GoogleNet [34] and ResNet [35].

The appearance-based technique was employed to define the ocular state in this investigation. Deep learning algorithms automatically extract features from images to establish and maintain representative features for a task, can disclose intricate details that are imperceptible to the naked eye, and can learn eye features according to different settings. As a result, deep learning has shown to be a viable method for recognizing eye states. Furthermore, unlike other machine learning methods, it delivers the system a favorable position in terms of speed and ease of use. Due to a paucity of data to reflect the status of the eyes under diverse environmental situations, CNN approaches based on the transfer learning have been devised.

In CEW [12] and ZJU [36] datasets, which are widely used in the literature for eye state recognition, classifications of previously widely used and pre-trained CNN models on the eye state recognition task were generated through transfer learning, and their results obtained were compared. On the both datasets, the most robust and high-performance pre-trained CNN model was handpicked in virtue of these comparisons. Modifying AlexNet, one of these pre-trained CNN networks, revealed the effectiveness and usefulness of the suggested new deep learning-based CNN model named DCNN on eye state recognition.

The main objectives of this work are as follows:

(1) Performing fine-tuning and evaluation of pre-trained CNN architectures on both eye state datasets.
(2) Building a new CNN architecture from AlexNet architecture, which is easy to use on many hardware and has a smaller depth compared to other pre-trained CNN architectures for eye state detection applications.
(3) Proposing a DCNN architecture based on AlexNet to increase eye state recognition accuracy.
(4) Evaluation of the proposed DCNN in CEW and ZJU eye state datasets, which are widely used in the literature, and comparison with state-of-art methods.
(5) Evaluation of the proposed method in a real-world scenario to demonstrate the reliability and robustness of eye state recognition in human–machine interaction.

The hereunder is how the rest of the article is organized. The flowchart, pre-trained CNN models, the suggested new CNN model, and the public blink dataset are all introduced in Sect. 2. The experiments used to validate the suggested method's performance are presented in Sect. 3. Section 4 is the discussion section that includes the comparison of the performance of the proposed method from the datasets with the state-of-art methods. The results and future research directions are covered in Sect. 5.

2 Materials and Methods

Considering a eye state dataset, a method based on DCNN was envisioned for automatic recognition of eye state (open or closed) in this research. The proposed approach is made up of four steps: (1) resizing the training and test images in the eye state dataset to make them acceptable for CNN model input, (2) training the pre-trained CNN models on the eye state dataset by adjusting hyperparameters, (3) measuring the performance of CNN models by evaluating the section reserved for testing on the eye state dataset on the created CNN models, and (4) then comparing the CNN models generated on eye state recognition to decide the most successful model. Figure 1 depicts the flow chart for the proposed scheme.

The images describing the eye state in this suggested method were retrieved from a eye state dataset. The full eye state dataset was first resized to make it appropriate for the input of pre-trained CNN architectures during the image preprocessing phase. Following that, the frequently used pre-trained CNN models (GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet, and DarkNet19) were trained on the eye state dataset at varied intervals to seek the most successful CNN model in eye state recognition. All these trained CNN models were evaluated on test data separated from the dataset, and their conclusions were calculated and compared. In the last round, AlexNet, one of the pre-trained CNN models was modified, and a new AlexNet-based CNN model was constructed. This CNN model was trained on the both eye state datasets before being evaluated and compared to earlier eye state recognition research.

2.1 Convolutional Neural Network

Artificial intelligence empowers machines to learn from their experiences, adapt to new inputs, and execute human-like activities with minimum human interaction. Artificial intelligence, in reality, is a branch of computer science that seeks to promote computers smarter. Artificial intelligence was first proposed in the 1950s, but due to the lack of processing power and machine storage at the time, it did not garner enough attention. However, due to the massive amount of data and breakthroughs in storage techniques, computer technology has recently made a comeback in the modern world [37]. Artificial intelligence involves sub-branches such as machine learning and deep learning. While machine learning functions in a single layer, deep learning operates in diverse layers at the same time. In order to construct

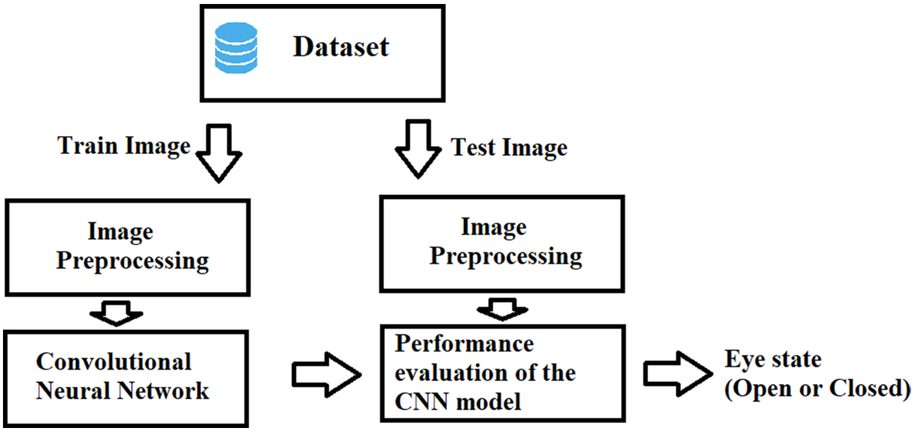
Dataset
Train Image
Test Image
Image Preprocessing
Image Preprocessing
Convolutional Neural Network
Performance evaluation of the CNN model
Eye state (Open or Closed)

Fig. 1 Flow chart of the proposed method

machine learning, the feature vector must also be retrieved. Feature vector extraction necessitates the use of experts in the domain. As a result, machine learning approaches cannot process raw data without the help of experts and preprocessing. By overcoming this challenge in machine learning, deep learning has made major progress. Deep learning was primarily employed in image analysis, sound analysis, robotics, autonomous vehicles, gene analysis, cancer diagnoses, and virtual reality after its inception [38]. Of the deep learning methods, CNN is the most well known and commonly utilized deep learning algorithm [33]. The fundamental advantage of CNN over machine learning methodologies is that it detects meaningful features from input without the need for human intervention [39]. CNNs have been used extensively and with excellent results in a variety of domains, including computer vision, speech processing, and face recognition, all of which entail exceedingly complex categorization problems [40]. You may need a lot of training data and a lot of skill to create a CNN from the ground up. For this reason, several researchers prefer to fine-tune CNN designs that have already been trained.

Numerous pre-trained CNN architectures have been designed and built on the ImageNet dataset by different researchers for multiple purposes: AlexNet, VGG16, GoogleNet, and so on [41]. Pre-trained CNN architectures respond differently depending on the dataset used for training and the creator's intended usage. The ImageNet dataset is commonly used to produce an architecture from scratch to tackle a problem since it is large enough to generate a good, generalized model. The parameters learned from the ImageNet dataset are transferred to the new architecture being built using the transfer learning method. Transfer learning optimizes the network's parameter training and tends to help to eliminate different sampling shortcomings in the dataset. In this study, GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet, and DarkNet19 architectures pre-trained from the ImageNet dataset were fine-tuned and utilized on the eye state dataset according to whether the eye is open or closed, and the guideline used to train these architectures could be seen in Fig. 2.

In eye state recognition, CNN models that have been pre-trained to execute a specific task may be converted to accomplish a new task using the transfer learning method (for example, as shown in Fig. 2). It is not always practicable to develop a CNN model from scratch because acquiring a large enough dataset can be troublesome. It is customary to employ pre-trained CNN models for specific applications under these circumstances (for example, the ImageNet dataset with 1.2 million images and 1000 categories) [32]. To begin, as illustrated in Fig. 2, the weights from the previously trained CNN architecture on the ImageNet dataset were transported to the newly built architecture via transfer learning, and the design was then fine-tuned to complement the eye state dataset.

One of the pre-trained CNN models utilized in the detection of eye state recognition, AlexNet was submitted by Krizhevsky et al. for the image classification task in the ImageNet Large-Scale Image Recognition Competition (ILSVRC-2012) [33]. This network has demonstrated that learning-based features may outperform hand-crafted features and disrupt previous computer vision trends. In terms of layers executed, the AlexNet structure encompasses a total of eight learnable layers, including five convolutional layers and three fully connected layers [41].

ResNet achieved first place in the classification assignment at the ILSVRC 2015 competition, which took place after AlexNet's triumph, with a 3.57% error rate on the ImageNet test set. Resnet is more comprehensive than AlexNet and VGG networks. Although ResNet possesses the deepest layers (152), it is less sophisticated than other pre-trained networks [35]. The ResNet architecture is available in 18, 34, 50, 101, and 152 layer versions. ResNet's 18-layer model is made use of to mitigate overfitting when the training dataset is not particularly large.

GoogleNet is a classic deep learning model offered by Szegedy et al. [34]. GoogleNet is a convolutional neural network with a depth of 22 layers. Unlike deeper networks, GoogleNet extracts more features and improves training results to achieve better training performance [34].

Sandler et al. [42] established MobileNetv2, a CNN architecture that strives to function well on mobile devices [42]. MobileNetv2 is founded on an inverted residual structure with residual connections between the bottleneck layers, unlike conventional CNN systems. In addition, as a nonlinear source, this structure seems to have an intermediary expansion layer that features somewhat deep curves to filter the derived features. There is an initial fully convolutional layer with 32 filters in the MobileNetv2 architecture, followed by 19 residual bottleneck layers [42].

Zhang et al. [43] developed ShuffleNet, a computationally cost-effective convolutional neural network designed exclusively for mobile devices. This novel architecture needs to employ two new processes as to substantially help lower computational cost while retaining accuracy: pointwise group convolution and channel shuffle [43].

DarkNet is a convolutional neural network that was created with the goal of being minimal and convenient. Diverse philosophies, including Network in Network, Inception, and Batch Normalization, are implemented to form this network [44]. In its structure, DarkNet19 emphasizes convolutional layers over fully connected layers. As a result, DarkNet19's structure comprises of 19 convolutional and five max pooling layers [45]. To cut down on the number of parameters, only (3×3) and (1×1) convolutional kernels are applied during training. Table 1 shows a comparison of the

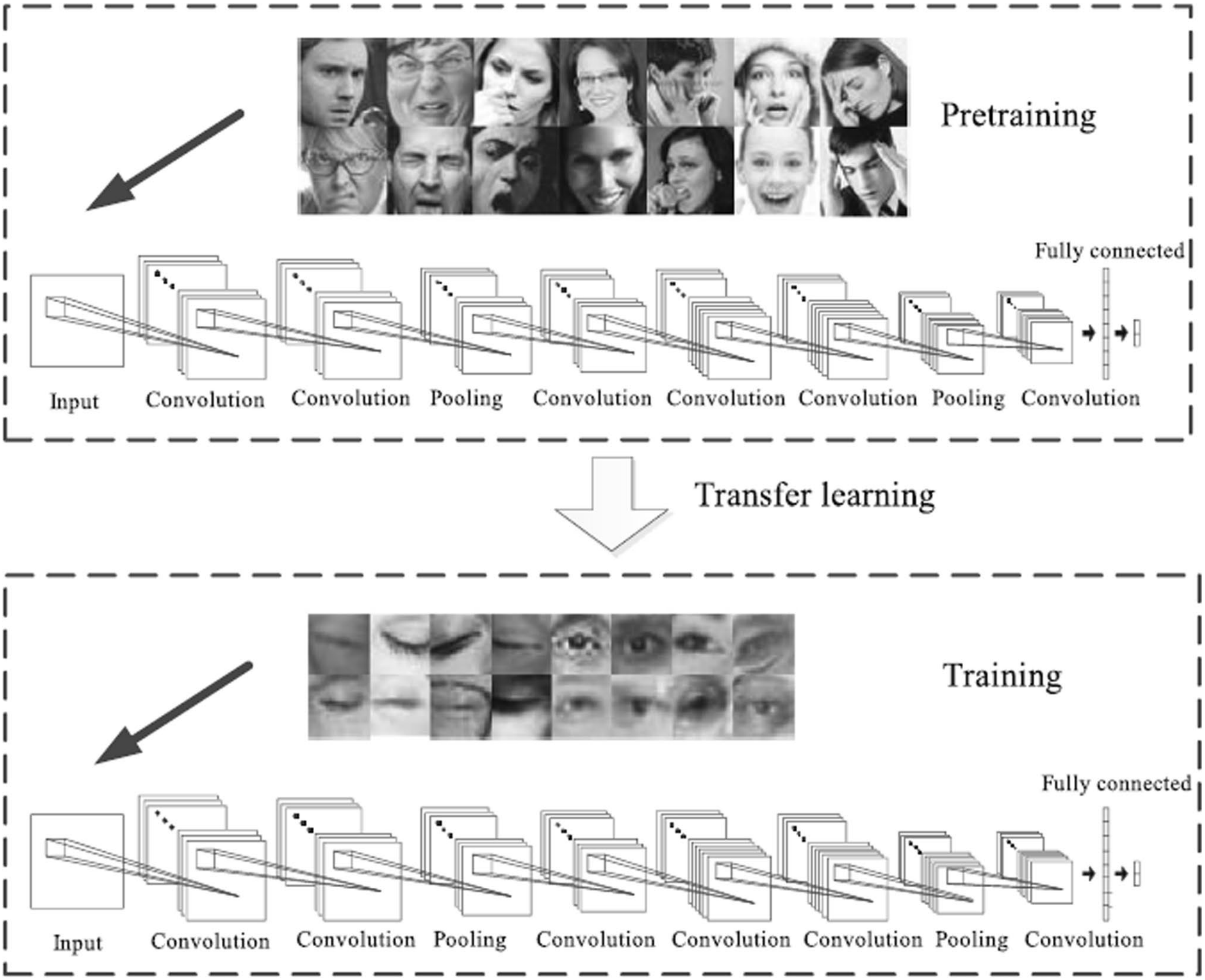

Fig. 2 Transfer learning framework for eye state recognition [26]

Table 1 Comparison of the characteristics of pre-trained networks

Network	Depth	Size (MB)	Parameters (millions)	Image input size
GoogleNet	22	27	7.0	224 × 224
ResNet18	18	44	11.7	224 × 224
MobileNetv2	53	13	3.5	224 × 224
ShuffleNet	50	5.4	1.4	224 × 224
AlexNet	8	227	61	227 × 227
DarkNet19	19	78	20.8	256 × 256

pre-trained CNN architectures employed in this work based on their characteristics.

According to Table 1, AlexNet seems to have the smallest depth, MobileNetv2 provides the smallest footprint on disk and main memory for the size of the network, and the AlexNet architecture offers the highest feature extraction (i.e., parameter) capability. If the network is saved to disk media, AlexNet uses a lot of space because it extracts relatively significantly more parameters. The image input size for GoogleNet, ResNet18, MobileNetv2, and ShuffleNet architectures is 224 × 224 pixels, while AlexNet performs slightly larger and DarkNet19 requires the largest image input size.

CNNs are optimized to continue operating with images, which distinguishes them from other methodologies. As a result, a 2D or 3D image is automatically considered to deliver the input of CNNs. Another distinction of CNN is that convolutional procedures are substantially deployed in its structure, as evidenced by the “convolutional” acronym of its name. The convolution layer, the pooling layer, and the fully connected layer are the three layers that make up a basic CNN structure. Subsampling layers such as normalization, activation, and pooling are utilized after the convolution layer.

The convolution layer is comprised of square number grids (kernels). To build and maintain the feature map, these cores conduct convolution with the layer's input. In other words, when extracting the feature map, the kernel runs the layer's input from left to right and bottom to top. The mathematical expression of the convolution operation, the convolution of a continuous function x and w $(x * w)$ (a) is defined in the following equation in all dimensions:

$$(x * w)(a) = \int x(t)w(a - t)da \tag{1}$$

Here, a is R^n for any $n \geq 1$. In addition, integral is replaced by the higher dimensional variant. However, the parameter t is assumed to be discrete, in practice, so discrete convolution is defined as seen in the following equation:

$$(x * w)(a) = \sum_{a} x(t)w(t - a) \tag{2}$$

where x is the input, w is the kernel, and the output is the feature map when a goes overall values in the input space. After the convolution layer, the pooling layer is essentially exploited. This layer's main objective is to try to reduce the image's size by combining particular areas of the image into a single value, and it also highlights the image's properties. Maximum and average pooling are two common pooling types in CNNs. Maximum pooling reports the maximum value, while average pooling estimates the average of nearby pixels. The activation function is another technique performed after the convolution layer. This function is used to incorporate non-linearity into deep learning models by teaching the non-linear prediction limits. The rectified linear unit (ReLU) is the most often utilized activation function in CNNs. The normalizing layer is one of the layers used after the convolution layer in CNN. This layer normalizes activations and gradients as they propagate through a network. Subsampling layers typically involve the ReLU, pooling, and normalization layers. One or more fully connected layers accompany the convolution and subsampling layers. Feature maps can now be entered into the ReLU or Softmax functions thanks to the completely connected layer. The softmax function then normalizes the fully connected layer's output. The categorization layer is the next and final layer. This layer allocates the class using the softmax function's outcome and also leverages the loss function to quantify the loss.

2.2 The Proposed AlexNet-Based DCNN

AlexNet is made up of five convolutional layers and three fully connected layers (FC6:4096, FC7:4096 and FC8:1000 neurons). AlexNet architecture was elected above other networks because it contains the smallest depth in the creation of the proposed new architecture. To do this, all layers were shifted to the newly constructed architecture using the transfer learning approach, with the exception of the FC8 fully connected layer, which AlexNet learnt from the ImageNet dataset. Then, according to the eye state dataset, a new randomly weighted fully connected layer with 1000 neurons was added to this design, followed by a fine-tuned fully connected layer to the two outputs (open or closed-eye state). AlexNet-based DCNN architecture was the term given to the newly constructed architecture. Figure 3 shows a representative implementation of the DCNN architecture. In the eye state recognition method, the proposed architecture was suggested as an alternative to pre-trained designs, and its performance was tested using two eye state datasets.

2.3 Image Processing

The image processing part shown in Fig. 1 allows the images in the dataset to be made in accordance with the entrance to the training of pre-trained CNN architectures. Resizing was applied according to the input layer size of these architectures. Considering the input dimensions of the pre-trained CNN architectures (Table 1), the training and test images in the eye state dataset were resized for GoogleNet [34], ResNet18 [35], MobileNetv2 [42] and ShuffleNet [43] (224×224), AlexNet [33] (227×227) and DarkNet19 [45] (256×256). The prepared eye state dataset was ready for the

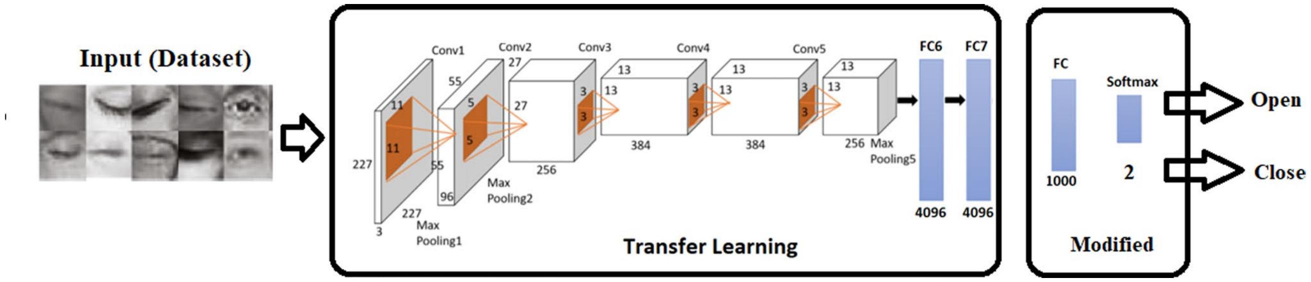

Fig. 3 DCNN architecture

training and testing phase as a result of this process. This process was used in two datasets according to the input of each trained CNN model.

2.4 Dataset

Pan et al. obtained the images from the ZJU dataset utilized in this investigation and they were under normal lighting and resolution conditions so that they were equivalent to real-world situations [36]. This dataset was used to detect eye state or differentiate between openness and closure of the eye, or to recognize blinking. The ZJU dataset was separated into two groups (training and testing) based on two categories (open and closed-eye images). The open and closed-eye images in this dataset are low-resolution, 24 × 24 pixels in size, and are also publicly available. Figure 4a exhibits examples of ocular images from the ZJU blink database, and Table 2 lists the database's specifications.

Another dataset used in this study for the performance evaluation of the proposed method in eye condition detection is the Closed Eyes in the Wild (CEW) [12] dataset. The CEW dataset has been collected in difficult variations caused by individual differences and different environmental changes such as light, blur and dark, known in real-world

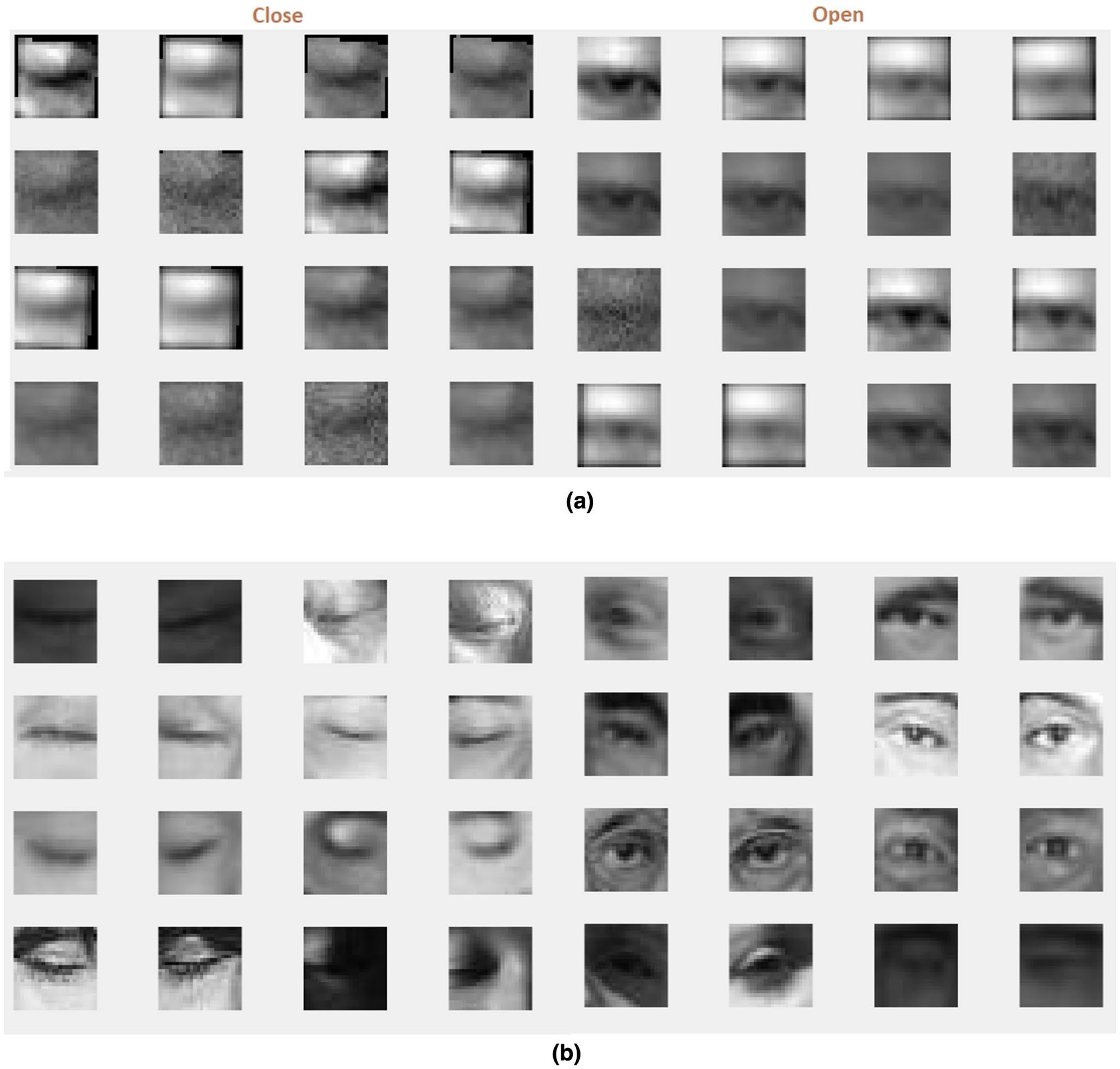

Fig. 4 Open and closed sample eye images in datasets: **a** ZJU and **b** CEW

Table 2 Details of the ZJU and CEW datasets

ZJU dataset			CEW dataset
Class	Training	Test	Image
Number of open eye images	5770	1230	2462
Number of close eye images	1574	410	2384
Total	7348	1640	4846

application scenarios for eye state detection. This dataset contains a set of human eye state images with a total of 2423 subjects selected from the Labeled Faces in the Wild database [46] of 1192 subjects with both eyes closed directly from the Internet and 1231 subjects with open eyes. Face detector and eye localization algorithm were applied to the images collected from the subjects, and finally, two 24 × 24 images of the eye region were obtained from each subject. Totally, the CEW dataset contains 2384 closed and 2462 open eye images. Details of this dataset are given in Table 2 and sample eye images are given in Fig. 4b.

One reason for choosing the ZJU dataset for the method proposed in this study is that it contains a good mix of low-resolution eye images. The CEW dataset, on the other hand, was chosen because it contains higher quality eye images obtained from various real-world environments. Therefore, the success of the proposed method on the task of eye state recognition was clearly demonstrated with this study, in which the training and verification of the method was carried out on two datasets with different characteristics. The dataset represented in the flowchart in Fig. 1 corresponds to the ZJU and CEW datasets used in this investigation.

2.5 Eye state Recognition Application

The issue of eye recognition detection, which is covered in this study, is likely to face real-world scenarios of various difficulties. For this reason, in addition to testing the proposed method on ZJU and CEW datasets, it has also been tested in a real-world scenario. For testing in real-world scenario, a video of a person in front of a digital screen was taken. In order for the proposed method to classify the eye state on the video, a flowchart given in Fig. 5 was created.

As seen in this flow chart, first, the face and eye regions must be determined. In this study, Viola-Jones detector [47] was used as the detection algorithm for real-time detection of face and eye regions, due to its high detection rate and real-time operation. This detector has been used in two different ways for detecting both face and eye regions. First, the image is taken via the web camera and then this image is fed to the face detector. The face detector detects the corresponding face area and provides it as an output. In the next step, the output is given to the eye detector. By way of the eye detector, two eye regions, right and left, are extracted from face region and then, the extracted eye regions are given as input to the DCNN architecture for eye state detection. The two eye regions are classified separately in the DCNN architecture. The two outputs obtained as a result of the classification are reported for evaluation. All these processes are continued until the images extracted from the video are finished.

3 Experiments and Results

On the ZJU and CEW eye state datasets, a strategy based on DCNN was presented for eye state recognition in this study (open or closed eye). The success of the proposed DCNN was first evaluated in the ZJU dataset, and then it was tested in another eye state recognition dataset, the CEW dataset. In the proposed scheme, first, pre-trained CNN architectures using the ImageNet dataset were exploited in the first stage to recognize eye states using the transfer learning method. The output of these architectures was scaled down to fit the ZJU dataset's number of classes, making it appropriate for training. The dimensions of the images used in the study were adjusted. The pre-trained CNN architectures were then trained and tested in a range of circumstances, and their results were measured. As a result, comparisons

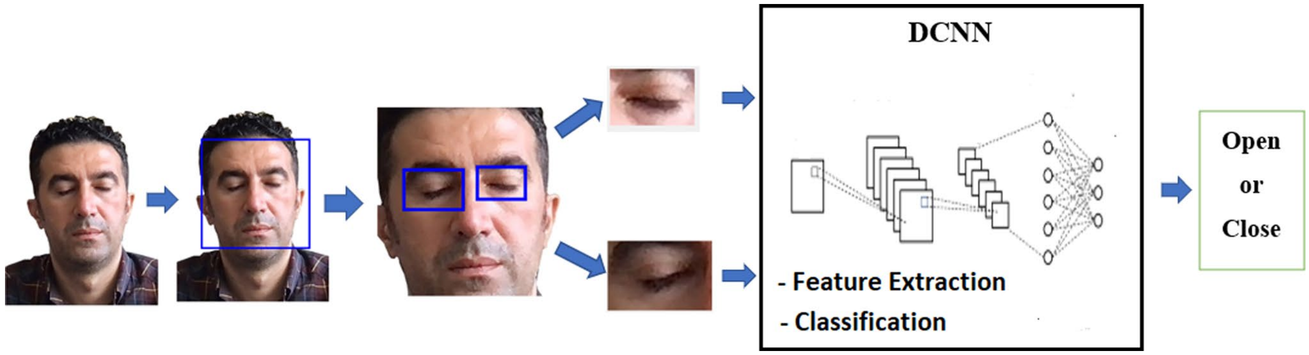

Fig. 5 Flowchart of eye state recognition application in real-world scenario

of pre-trained CNN and DCNN architectures on the ZJU dataset were carried out and the most successful and effective CNN model was determined in this study. AlexNet and DCNN, which achieved significant success in the ZJU dataset, were also retrained and evaluated in the CEW dataset. In addition, a video was taken from a real-world scenario to demonstrate the effectiveness of the proposed DCNN. On this video, the eye area of the person was extracted by the detection algorithm and the obtained right and left eye areas were classified in DCNN. The validation in this video was performed separately for DCNN trained on two datasets and the most reliable eye state recognition method was revealed by comparing the obtained findings.

Pre-trained CNN and DCNN architectures used in the proposed system flow were implemented on a computer with Windows 10 operating system and 16 GB RAM, Nvidia GTX 1650Ti and Intel Core i7 & 2.6 GHz in MATLAB 2020a environment. All used architectures were trained on the GPU (graphic processing unit) and their performances were evaluated.

3.1 Comparison of Traditional Pre-trained CNN Architectures on ZJU Dataset

In the MATLAB environment, the weights of the GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet, and DarkNet19 architectures learnt on the ImageNet dataset were registered. As a result, the weights of these architectures for eye state recognition were fine-tuned through the ZJU dataset and then imported using the transfer learning method (as seen in Fig. 2). After the weights of the pre-trained CNN architectures were transferred by transfer learning, the dimensions of the eye images in the ZJU dataset were resized for GoogleNet, ResNet18, MobileNetv2 and ShuffleNet (224 × 224), AlexNet (227 × 227) and DarkNet19 (256 × 256) and were made suitable for the input layer of the architectures.

One of the most crucial aspects of training CNNs is the hyper-parameter optimization. The same hyperparameters were utilized for all the designs in this research after transfer learning was applied to GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet, and DarkNet19, and the training was started. Details of the hyperparameters picked for these architectures are given in Table 3.

Table 3 Details of the hyperparameters

Parameter	Value/type
Loss criterion	Crossentropyex
Optimizer	Adam
Learning rate	0.0001
L2 regularization	0.0001
Batch size	32
Gradient decay factor	0.9
Epoch	20
Network output	Open Close

The weights of GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet and DarkNet19 architectures transferred by the transfer learning were updated with the backpropagation algorithm to the new weight and bias values during training in the ZJU dataset. Later, the Softmax function was deployed to apply the fully connected layer's input data to the network output. The Softmax function made the input data normalized so that the sum of the values equaled 1. The weights of the network were also modified during the training of pre-trained CNN architectures, and the cross-entropy loss function was utilized to endeavor to bring the margin of error closer to 0. The loss function ensures that the network's output is as close to the desired output as possible (accuracy values). The mathematical expression of cross entropies used in the classification layer of all CNN architectures used in this study is given in the following equation:

$$H(p,q) = -\sum_{x} p(x)\log q(x) \tag{3}$$

In Eq. (3), x is the number of classes; q is the output of the softmax function; p represents the categorical class output. The functions utilized in CNN architectures' structures may cause non-linear values during training on the ZJU dataset. Thus, the Adam [48] optimizer was adopted in CNN architectures to decrease the discrepancy between the original and output values. Adam is an optimization technique that combines the best elements of the AdaGrad and RMSProp algorithms to produce a solution that can handle sparse gradients in noisy applications. It is a prominent method in the field of deep learning since it yields successful outcomes speedily. The optimization technique is also a hyper-parameter and Adam optimizer was used in all trained CNN architectures in this study.

GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet and DarkNet19 architectures were tested with images in the section dedicated to testing of the ZJU dataset to determine their performance after the training was completed. Next, a confusion matrix was created according to the test results of each architecture in order to measure the performance of these architectures. This matrix was used to compute the most popular performance criteria accuracy (Eq. 4), specificity (Eq. 5), sensitivity (Eq. 6), precision (Eq. 7) and $F1$ score (Eq. 8) metrics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$F_1\text{score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{8}$$

The TP (true positive) amount indicates that an eye state is recognized as closed when it is closed, whereas the TN (true negative) value indicates that an eye state is clearly recognized as open when it is open. FP (false positive) refers to the recognition of an open eye state as closed, while FN (false negative) refers to the perception of a closed eye state as open. The area under the receiver-operating characteristic (ROC) curve is one of the most critical performance indicators used in classifier performance evaluation as AUC. The stronger the classifier, the greater the AUC score. The AUC value is a graph that summarizes the classifier's performance across all conceivable values. The performance criteria and AUC values gained from the confusion matrices formed with the help of testing GoogleNet, ResNet18, MobileNetv2, ShuffleNet, AlexNet and DarkNet19 architectures on the ZJU dataset are given in Table 4.

In the comparison of architectures on the ZJU dataset in Table 4, AlexNet performed best in the accuracy measure, whereas ResNet18 performed best in the sensitivity. The specificity metric, on the other hand, was best performed by AlexNet and MobileNetv2. ShuffleNet also outperformed in the precision metric, AlexNet outperformed on the $F1$ score metric, and MobileNetv2 outperformed in the AUC metric. In the performance metrics mentioned in the confusion matrix calculation, AlexNet scored the best performance on three of the six metrics, according to the comparative results (accuracy, specificity, and $F1$ score). Despite falling behind in sensitivity, precision, and AUC, AlexNet outperformed other methodologies.

3.2 Proposed Method vs. AlexNet on ZJU Dataset

Certain adjustments were introduced to the last fully connected layer structure of the AlexNet architecture, which is the finest performing of the pre-trained CNN architectures, so as to maximize the accuracy of eye state recognition on the ZJU dataset, and a new DCNN architecture was constructed. The weights of the other layers were transported through using the transfer learning approach after the FC8 fully connected layer of AlexNet was extracted to the DCNN architecture. A new fully connected layer with random weight was implemented in place of the extracted FC8 layer, and then a DCNN architecture was established by modifying the network's output according to the ZJU dataset. Based on DCNN's input layer size, the images in the ZJU dataset are resized to 227 × 227. Next, DCNN was trained by adjusting the hyperparameters in Table 3. The confusion matrix was established to quantify the DCNN architecture's performance. The performance metrics of the DCNN architecture were calculated using the information gathered from the confusion matrix and their comparison with AlexNet is given in Table 5.

Compared to AlexNet, DCNN showed superior performance in all metrics except for the sensitivity in the

Table 4 Performance comparison of pre-trained CNN architectures on ZJU dataset

Method	Accuracy	Sensitivity	Specificity	Precision	$F1$ score	AUC
GoogleNet	0.9055	0.9756	0.8821	0.7339	0.8377	0.98668
ResNet18	0.9024	**0.9780**	0.8772	0.7264	0.8337	0.98867
MobileNetv2	0.9537	0.9415	**0.9577**	0.8813	0.9104	**0.99033**
ShuffleNet	0.9573	0.9220	0.9691	**0.9087**	0.9153	0.98440
AlexNet	**0.9579**	0.9585	**0.9577**	0.8831	**0.9193**	0.98783
DarkNet19	0.8439	0.6171	0.9195	0.7188	0.6640	0.91843
Average	0.9201	0.8988	0.9272	0.8087	0.8467	0.97606

The values indicated in bold in the columns represent the highest results

Table 5 Performance comparison of DCNN and AlexNet architectures on ZJU dataset

Method	Accuracy	Sensitivity	Specificity	Precision	$F1$ score	AUC
AlexNet	0.9579	**0.9585**	0.9577	0.8831	0.9193	0.98783
DCNN	**0.9732**	0.9537	**0.9797**	**0.9399**	**0.9467**	**0.99365**

The values indicated in bold in the columns represent the highest results

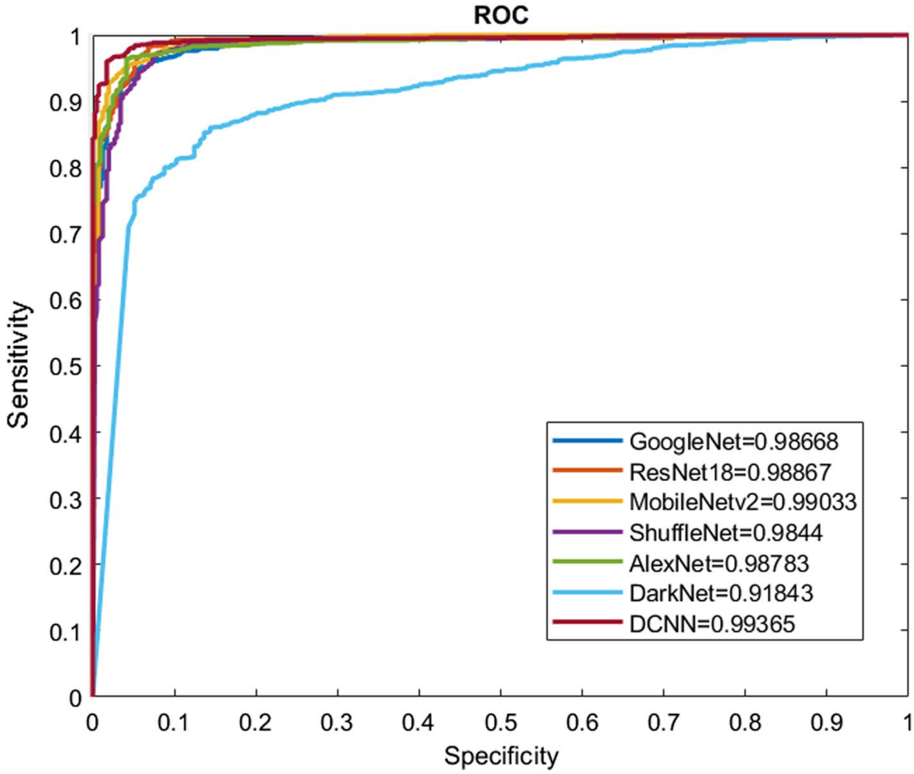

Fig. 6 ROC curves of CNN architectures

Table 6 Performance comparison of DCNN and AlexNet architectures on CEW dataset

Method	Accuracy	Sensitivity	Specificity	Precision	*F*1 score	AUC
AlexNet	97.11	**98.74**	95.53	95.53	97.11	99.42
DCNN	**97.93**	**98.74**	**97.15**	**97.11**	**97.92**	**99.69**

The values indicated in bold in the columns represent the highest results

calculated performance metrics. The ROC curve has been generated and shown in Fig. 6 for a summary representation of performance comparisons with DCNN compared to pre-trained CNN architectures.

As seen in the ROC curve in Fig. 6, DCNN performed the best distinguishing between the two classes in eye state recognition and achieved the best classification ability compared to other CNN architectures.

3.3 Proposed Method vs. AlexNet on CEW Dataset

As seen in Table 5, DCNN and AlexNet architectures are the two architectures with the highest success in the ZJU dataset. CEW dataset was used to test the performance of these two architectures. The eye state images in the CEW dataset are split by 90% for training and 10% for testing. Afterwards, image preprocessing was applied on the CEW dataset and it was made suitable for DCNN and AlexNet architectures inputs. In the following step, the hyperparameters given in Table 3 were adjusted for training DCNN and AlexNet on the CEW dataset. Of these hyperparameters, only the epoch hyperparameter was set at 40 by increasing the epoch number to solve the overfitting problem, since

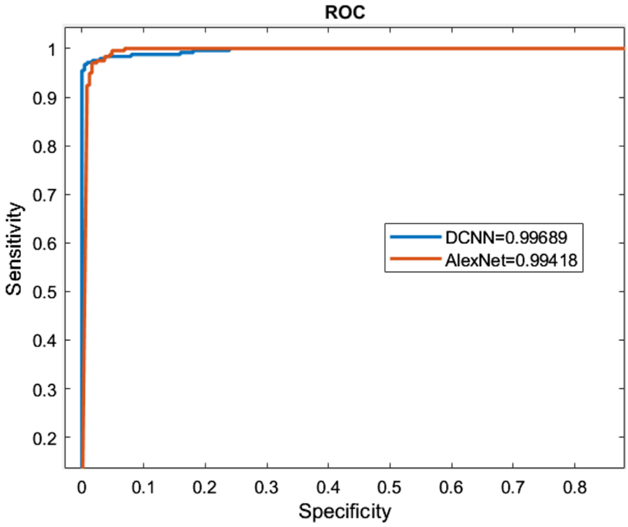

Fig. 7 ROC curves of DCNN and AlexNet architectures

CEW has approximately 2 times less data than ZJU. Then, AlexNet and DCNN architectures were trained using the training data in the CEW dataset. The performance metrics obtained after performing the test operation on the test data are given in Table 6. The ROC curve representing the performance summary of AlexNet and DCNN architectures in the CEW dataset is also given in Fig. 7.

3.4 Real-World Scenario Testing of the Proposed Method

In this study, the proposed method was also tested in a real-world scenario. To test the proposed method in a real-world scenario, a 1-min video of a person looking at a digital screen was taken. This video has been analyzed in the eye state detection application detailed in Sect. 2.5. In this

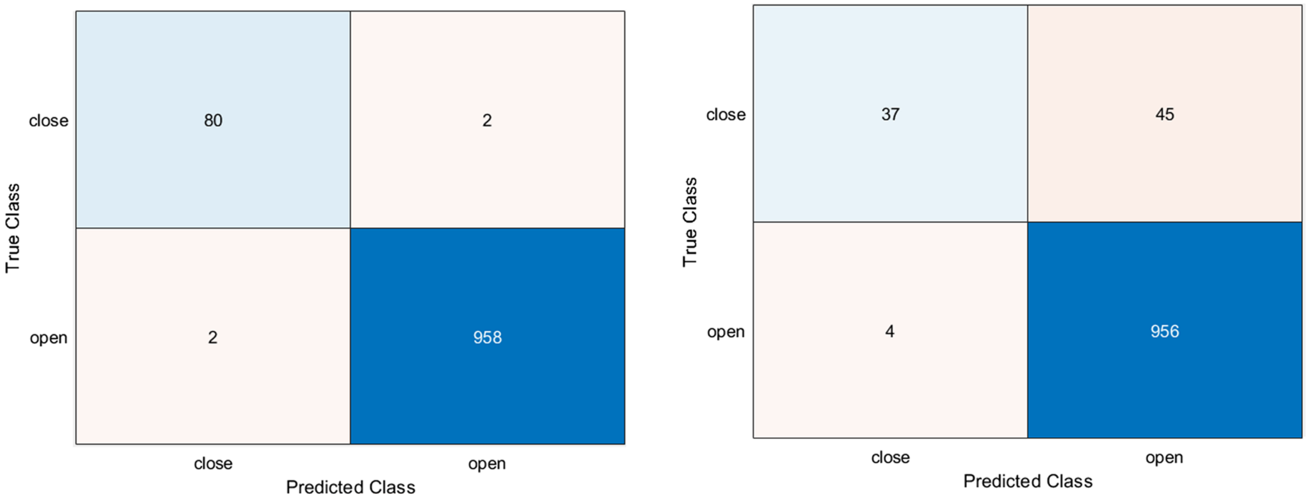

Fig. 8 Confusion matrices from real scenario for DCNN: **a** trained in CEW dataset, **b** trained in ZJU dataset

Table 7 Performance comparison of DCNN (trained ZJU and CEW datasets)

Method	Accuracy	Sensitivity	Specificity	Precision	*F*1 score
DCNN (trained ZJU)	95.30	45.12	99.58	90.24	60.16
DCNN (trained CEW)	**99.62**	**97.56**	**99.79**	**97.56**	**97.56**

The values indicated in bold in the columns represent the highest results

Table 8 Comparison of DCNN architecture and previous studies on ZJU dataset

Source	Method	Accuracy	AUC
Pauly and Sankar [18]	HOG + SVM	85.62	–
Pauly and Sankar [23]	HOG + SVM	85	–
Lee et al. [19]	SVM	95.14	–
Zhao et al. [22]	TRL + DA	97.19	99.29
Song et al. [12]	MultiHPOG + LTP + Gabor + SVM	96.83	99.27
Wu et al. [1]	LBP + SVM	90.37	–
Liu et al. [25]	Gabor + LBP + HOG + SVM	95.42	98.02
Eddine et al. [21]	Multi-TPLBP + SVM	95.18	97.83
Dong et al. [11]	HOG + Random Ferns	94.76	98.89
Liu et al. [26]	WBCNNTL	97.20	–
Saurav et al. [28]	DCNNE	**97.99**	–
Liang et al. [29]	DFNN	96.96	99.03
Wang et al. [27]	HOG + Autoencoder	94.75	98,20
Proposed method	DCNN	97.32	**99.37**

The values indicated in bold in the columns represent the highest results application, first, images were extracted from the video and the extracted images were given to the face detector. The face detector obtained 521 images in which it detected the presence of the face from the images in the video. Then, by applying the eye detector to 521 images, 1042 eye regions were extracted and were classified in the proposed DCNN architecture. In addition, the eye state recognition application was run twice for both DCNN trained on ZJU dataset and DCNN trained on CEW dataset. Confusion matrices created by DCNNs result from these two evaluations and are presented in Fig. 8. The performance metrics calculated from the confusion matrices are also given in Table 7.

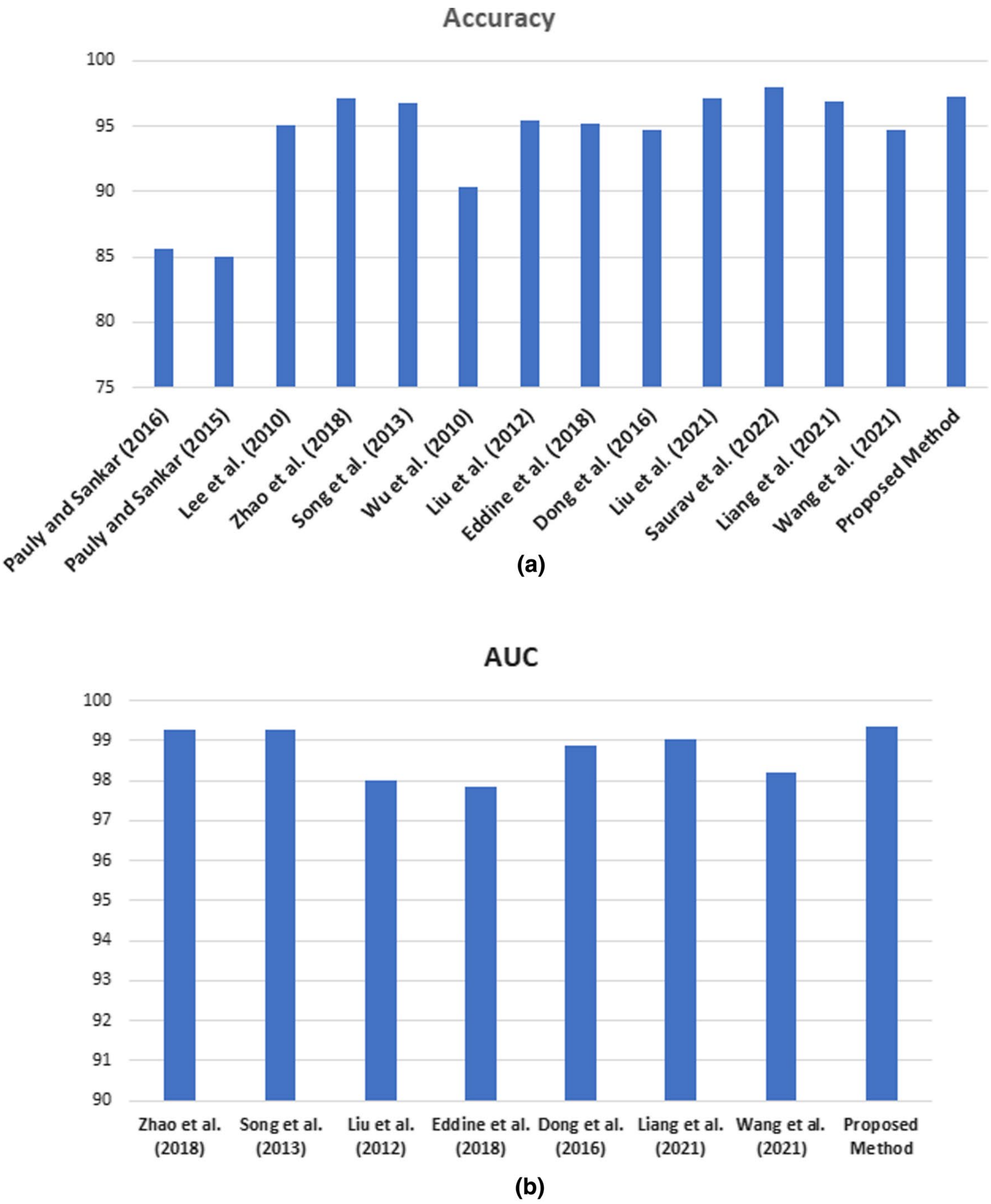

Fig. 9 Comparison of performance metrics by previous studies on the ZJU dataset: a accuracy and b AUC

4 Discussion

Eye state recognition directs developments in many fields, especially HMI. Recently, interest in deep learning-based CNN, which is widely used in many different tasks, has also increased in eye state recognition. In this study, performance metrics of the proposed DCNN architecture were compared with previous studies on the ZJU dataset and are given in Table 8.

It is worth noting that the performance metrics like accuracy and AUC were usually shared in the earlier studies when evaluating the methods on the ZJU dataset. Therefore, DCNN architecture was compared in Table 8, considering accuracy and AUC values. In the compared performance metrics, the DCNN method has shown that it is one of the most successful methods in AUC performance metric for eye state recognition compared to previous studies. As for the accuracy metric, the DCNNE suggested by Sauraw et al. [28] has the highest accuracy, but the AUC metric was not presented. Therefore, it has been observed that the proposed method has the highest success in comparing the state-of-the-art methods in the literature with both AUC and accuracy metrics presented together. The closest success to the proposed method in both parameters is Zhao et al. [22].

The studies with the closest accuracy value to the proposed DCNN architecture were the studies of Liu et al. [26], Zhao et al. [22] and Liang et al. [29], as seen in Table 8. CNN based on deep learning, as in the proposed DCNN architecture, was one of the strategies used in these investigations. Machine learning algorithms were commonly utilized on the ZJU dataset till 2018, as shown in Table 8. However, it has lately been observed that CNN-based technologies, including deep learning, are being applied. A comparative graph of the accuracy and AUC values of the DCNN method with previous studies on the ZJU dataset has been given in Fig. 9.

When the studies with the ZJU dataset are examined, it is seen that machine learning classifiers such as SVM are used mostly after algorithms that extract handmade features in eye state detection. Eye state recognition leads to advances in many areas, especially HMI. When the studies with the ZJU dataset are examined, it is seen that machine learning classifiers, such as SVM, have been used after algorithms that extract handmade features in eye state detection.

The performance of DCNN is also evaluated on the CEW dataset. The comparison of the accuracy and AUC values obtained on the CEW dataset of the proposed method with the state-of-art methods is given in Table 9. A comparative graph of the accuracy and AUC values of the DCNN method with previous studies on the CEW dataset is also given in Fig. 10.

Table 9 Comparison of DCNN and AlexNet architectures and previous studies on CEW dataset

Source	Method	Accuracy	AUC
Zhao et al. [22]	DINN	97.19	99.57
Sauraw et al. [28]	DCNNE	97.50	–
Liang et al. [29]	DFNN	97.37	**99.71**
Song et al. [12]	Projection	70.10	–
Song et al. [12]	MultiHPOG-LTP + Gabor + SVM	94.72	95.19
Liu et al. [26]	WBCNNTL	97.40	–
Our study	AlexNet	97.11	99.42
	DCNN	**97.93**	99.69

The values indicated in bold in the columns represent the highest results

As seen in Table 9 and Fig. 10, DCNN achieved the best performance in the accuracy metric. In the AUC value of the proposed method, it took the second place after Liang et al. [29] with a slight difference. The DFNN method proposed by Liang et al. evaluated in both CEW and ZJU datasets as given in Tables 8 and 9. In the performance comparison of DFNN with the proposed method on the ZJU dataset, the proposed method outperformed DFNN in both accuracy and AUC metric. In CEW dataset, Saurav et al. [28] and Liu et al. [26] are the closest to the accuracy of the proposed method. As a result of the comparative analysis of the proposed method on the CEW dataset, it is seen that it has achieved superior performance than in previous studies.

When the results obtained from the ZJU dataset and the CEW dataset are examined, it is seen that the proposed DCNN architecture can be used successfully for eye state detection. Due to this, the use of CNNs in eye state recognition has become remarkable in terms of both ease of use and improved success thanks to automatic feature extraction. Therefore, the use of CNN-based approaches has become an important tool in tasks where eye state recognition is important.

Eye state recognition is not easy, as many real-world scenarios are faced. Accordingly, the proposed method was tested by taking a video from a real-world environment. Test results of the proposed method trained on CEW and ZJU datasets are presented in Table 7 and Fig. 8. Of the proposed method, the one trained with ZJU achieved over 95% performance and the one trained with CEW achieved over 99% performance. In the confusion matrices created as a result of these tests, it was seen that the proposed method trained with CEW showed the best performance in all performance metrics compared to the one trained with ZJU. Comparative analyzes have shown that the proposed method has effective and reliable performance in real-world scenarios and is an

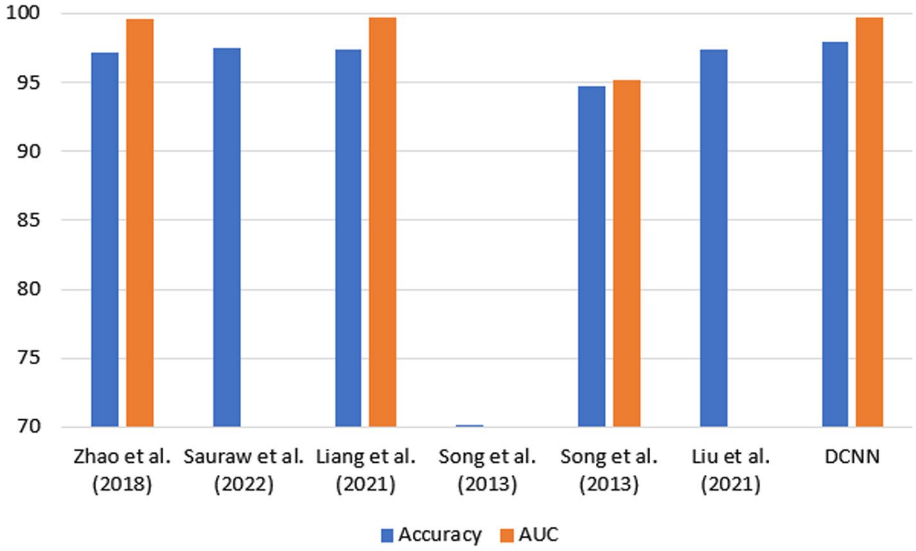

Fig. 10 Comparison of performance metrics according to accuracy and AUC values by previous studies on the CEW dataset

alternative method that can be used in applications developed for eye state detection.

5 Conclusion

Eye state recognition has a broad array of applications, from HMI systems to monitoring driver fatigue, dry eye, and computer vision syndrome associated with continuous use of digital screens. Recognition of eye state, whether sensitively on or off, can pave the path for the creation of innumerable technologies in this area. Using the ZJU and CEW datasets, a strategy based on DCNN was presented for eye state recognition in this study. In light of the findings, the performances of pre-trained CNN architectures trained on the ZJU dataset were compared, and AlexNet was proven to cause the best performance. Modifications were implemented to the AlexNet structure in order to maximize the likelihood of success on the ZJU dataset. The performance of the developed DCNN architecture was measured on both the ZJU and CEW datasets. The developed DCNN architecture outperformed the CNN architectures utilized in the study and demonstrated the greatest performance, according to the comparative data in ZJU and CEW datasets (Tables 5 and 6). The obtained results from the DCNN architecture were compared to those of prior studies on the both datasets. When comparing the accuracy and AUC performance metrics given in prior studies, the DCNN architecture was showcased to be the most efficient strategy for eye state recognition in the literature. The fact that DCNN outperforms machine learning techniques in prior experiments has piqued interest in deep learning, particularly CNN, for recognizing eye states. In addition, in this study, the proposed method has also been tested in a real-world scenario, and the results have shown that this method has effective performance even in various scenarios. Future studies should incorporate data augmentation approaches or combine datasets to expand the number of open or closed eye images in order to improve the performance of the approaches. In addition, we think that the data augmentation-based CNN method can give more successful results in eye state recognition.

Acknowledgements The authors would like to thank the editor and anonymous referees of this manuscript.

Author Contributions Methodology: GEG and IK; conceptualization: GEG and IK; software: IK; supervision: GEG and UE; validation: GEG and UE; writing—original draft: IK and NOS; writing—review and editing: GEG and UE. All the authors have read and agreed to the final version of the manuscript.

Funding Not applicable.

Availability of Data and Materials The ZJU and CEW datasets used in this study are open access (available at http://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html).

Declarations

Conflict of Interest The authors declare that they have no conflicts of interest to report regarding the present study.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Wu, Y.S., Lee, T.W., Wu, Q.Z., Liu, H.S.: An eye state recognition method for drowsiness detection. In: 2010 71st Vehicular Technology Conference, pp. 1–5 (2010)
2. Meshach, W.T., Hemajothi, S., Anita, E.M.: Real-time facial expression recognition for affect identification using multi-dimensional SVM. J. Ambient. Intell. Humaniz. Comput. **12**(6), 6355–6365 (2021)
3. Ruan, D., Yan, Y., Lai, S., Chai, Z., Shen, C., Wang, H.: Feature decomposition and reconstruction learning for effective facial expression recognition. In: 2021 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7660–7669 (2021)
4. Nanthini, N., Puviarasan, N., Aruna, P.: Eye blink-based liveness detection using odd kernel matrix in convolutional neural networks. In: International Conference on Innovative Computing and Communications, pp. 473–483 (2022)
5. Kuwahara, A., Hirakawa, R., Kawano, H., Nakashi, K., Nakatoh, Y.: Eye fatigue prediction system using blink detection based on eye ımage. In: 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–3 (2021)
6. Cyganek, B., Gruszczyński, S.: Hybrid computer vision system for drivers' eye recognition and fatigue monitoring. Neurocomputing **126**, 78–94 (2014)
7. Blehm, C., Vishnu, S., Khattak, A., Mitra, S., Yee, R.W.: Computer vision syndrome: a review. Surv. Ophthalmol. **50**(3), 253–262 (2005)
8. Divjak, M., Bischof, H.: Eye blink based fatigue detection for prevention of computer vision syndrome. In: IAPR Conference on Machine Vision Applications, pp. 350–353 (2009)
9. Królak, A., Strumiłło, P.: Eye-blink detection system for human–computer interaction. Univ. Access Inf. Soc. **11**(4), 409–419 (2012)
10. Huang, R., Wang, Y., Guo, L.: P-FDCN based eye state analysis for fatigue detection. In: 2018 IEEE 18th International Conference on Communication Technology (ICCT), pp. 1174–1178 (2018)
11. Dong, Y., Zhang, Y., Yue, J., Hu, Z.: Comparison of random forest, random ferns and support vector machine for eye state classification. Multimed. Tools Appl. **75**(19), 11763–11783 (2016)
12. Song, F., Tan, X., Liu, X., Chen, S.: Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. Pattern Recogn. **47**(9), 2825–2838 (2014)
13. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
15. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
16. Tian, Y.L., Kanade, T., Cohn, J.F.: Eye-state action unit detection by gabor wavelets. In: International Conference on Multimodal Interfaces, pp. 143–150 (2000)
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
18. Pauly, L., Sankar, D.: Non intrusive eye blink detection from low resolution images using HOG-SVM classifier. Int. J. Image Graph. Signal Process. **8**(10), 11 (2016)
19. Lee, W.O., Lee, E.C., Park, K.R.: Blink detection robust to various facial poses. J. Neurosci. Methods **193**(2), 356–372 (2010)
20. Radlak, K., Smolka, B.: A novel approach to the eye movement analysis using a high speed camera. In: 2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pp. 145–150 (2012)
21. Eddine, B.D., Dos Santos, F.N., Boulebtateche, B., Bensaoula, S.: Eyelsd a robust approach for eye localization and state detection. J. Signal Process. Syst. **90**(1), 99–125 (2018)
22. Zhao, L., Wang, Z., Zhang, G., Qi, Y., Wang, X.: Eye state recognition based on deep integrated neural network and transfer learning. Multimed. Tools Appl. **77**(15), 19415–19438 (2018)
23. Pauly, L., Sankar, D.: A novel method for eye tracking and blink detection in video frames. In: 2015 International Conference on Computer Graphics, Vision and Information Security (CGVIS), pp. 252–257 (2015)
24. Han, Y.J., Kim, W., Park, J.S.: Efficient eye-blinking detection on smartphones: A hybrid approach based on deep learning. Mob. Inf. Syst. (2018). https://doi.org/10.1155/2018/6929762
25. Liu, X., Tan, X., Chen, S.: Eyes closeness detection using appearance based methods. In: International Conference on Intelligent Information Processing, pp. 398–408 (2012)
26. Liu, Z.T., Jiang, C.S., Li, S.H., Wu, M., Cao, W.H., Hao, M.: Eye state detection based on weight binarization convolution neural network and transfer learning. Appl. Soft Comput. **109**, 107565 (2021)
27. Wang, H., Li, B., Shic, F., Hu, B., Wang, Q.: Comparing robustness and efficiency of closed eye detection in ımages. In: 6th International Conference on Image, Vision and Computing (ICIVC), pp. 6–12 (2021)
28. Saurav, S., Gidde, P., Saini, R., Singh, S.: Real-time eye state recognition using dual convolutional neural network ensemble. J. Real-Time Image Proc. (2022). https://doi.org/10.1007/s11554-022-01211-5
29. Liang, H., Liu, C., Chen, K., Kong, J., Han, Q., Zhao, T.: Controller fatigue state detection based on ES-DFNN. Aerospace **8**(12), 383 (2021)
30. Arshad, M., Qureshi, M., Inam, O., Omer, H.: Transfer learning in deep neural network based under-sampled MR image reconstruction. Magn. Reson. Imaging **76**, 96–107 (2021)
31. Zhang, Y., An, M.: Deep learning-and transfer learning-based super resolution reconstruction from single medical image. J. Healthc. Eng. (2017). https://doi.org/10.1155/2017/5859727
32. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
33. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural İnf. Process. Syst. **25**, 1097–1105 (2012)
34. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
35. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
36. Pan, G., Sun, L., Wu, Z., Lao, S.: Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: 2007 11th International Conference on Computer Vision, pp. 1–8 (2007)
37. Lalle, Y., Fourati, M., Fourati, L.C., Barraca, J.P.: Communication technologies for Smart Water Grid applications: Overview, opportunities, and research directions. Comput. Netw. (2021). https://doi.org/10.1016/j.comnet.2021.107940
38. Özkan, I., Ulker, E.: Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. Gaziosmanpaşa Bilimsel Araştırma Dergisi **6**(3), 85–104 (2017)

39. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognit. **77**, 354–377 (2018)
40. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Farhan, L.: Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. J. Big Data **8**(1), 1–74 (2021)
41. Srinivas, S., Sarvadevabhatla, R.K., Mopuri, K.R., Prabhu, N., Kruthiventi, S.S., Babu, R.V.: A taxonomy of deep convolutional neural nets for computer vision. Front. Robot. AI **2**, 36 (2016)
42. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
43. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
44. Amjoud, A. B., Amrouch, M.: Convolutional neural networks backbones for object detection. In: 2020 International Conference on Image and Signal Processing, pp. 282–289 (2020)
45. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
46. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments, pp. 07–49. University of Massachusetts, Amherst (2007)
47. Deshpande, N.T., Ravishankar, S.: Face detection and recognition using viola-jones algorithm and fusion of PCA and ANN. Adv. Comput. Sci. Technol. **10**(5), 1173–1189 (2017)
48. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)