Applied Soft Computing 53 (2017) 168–180

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Face detection and recognition in an unconstrained environment for mobile visual assistive system

Shonal Chaudhry*, Rohitash Chandra

*Artificial Intelligence and Cybernetics Research Group, Software Foundation, Nausori, Fiji*[1]

ARTICLE INFO

*Article history:*
Received 14 August 2015
Received in revised form 1 September 2016
Accepted 15 December 2016
Available online 26 December 2016

*Keywords:*
Assistive system
Computer vision
Face detection
Face recognition
Mobile computing

ABSTRACT

We present a visual assistive system that features mobile face detection and recognition in an unconstrained environment from a mobile source using convolutional neural networks. The goal of the system is to effectively detect individuals that approach facing towards the person equipped with the system. We find that face detection and recognition becomes a very difficult task due to the movement of the user which causes camera shakes resulting in motion blur and noise in the input for the visual assistive system. Due to the shortage of related datasets, we create a dataset of videos captured from a mobile source that features motion blur and noise from camera shakes. This makes the application a very challenging aspect of face detection and recognition in unconstrained environments. The performance of the convolutional neural network is further compared with a cascade classifier. The results show promising performance in daylight and artificial lighting conditions while the challenges lie for moonlight conditions with the need for reduction of false positives in order to develop a robust system. We also provide a framework for implementation of the system with smartphones and wearable devices for video input and auditory notification from the system to guide the visually impaired.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Computer vision algorithms execute some of the most computational intensive tasks in problems such as pattern recognition and motion analysis [1]. The simple object detection algorithms [2,3] require a significant amount of computational power due to the amount of data that needs to be processed in large-scale applications. Modern desktop computers are able to execute these applications in real-time, howsoever, the challenge is for mobile applications to handle computationally intensive tasks that produce heat and rapidly consume battery power. Modern computers are able to execute these programs in real-time without any major issues whereas the challenge is for mobile applications due to limitations in battery power and heavy computation that creates heat. Mobile devices can harness the full power of real-world computer vision applications when applications are built taking into account the limitations that are faced by them [4].

Mobile object detection systems have a wide range of applications due to their portability [5,6]. While detection of static objects in general is a relatively easier task, the detection of moving objects is more challenging [1]. Some of the examples of mobile object detection are assistive systems for disabled persons [6] and iris recognition systems [5]. The inclusion of motion in computer vision applications incorporates major difficulties which can include blur, constant scale and position changes, obstructions, and illumination changes [3]. Advanced detection methods such as neural networks are required to account for these challenges with the hope to achieve satisfactory performance [7,8]. The *SmartVision* prototype [9] is an example of a mobile-based assistive system that provides navigation for disabled persons. It used a combination of computer vision, geographic information system and global positioning system for object, obstacle and path detection. Moreover, Willis et al. presented a mobile-based assistive system that allowed users to navigate an environment using a radio frequency Identification (RFID) tag grid [10]. This tag grid had RFID tags programmed with coordinates and descriptions of the surroundings for providing navigation to users. Furthermore, a mobile iris recognition system has been presented where the system provided pupil and iris segmentation with a detection rate of 99% [11].

Neural networks consist of interconnected processors called neurons which are loosely modelled after biological neurons [8]. Convolutional neural networks (CNNs) are specialised neural networks that are primarily designed for image recognition tasks [12]. Some of these include face detection, expression recognition, object detection and object recognition [13–15].

* Corresponding author.
*E-mail address:* shonal_c@outlook.com (S. Chaudhry).
[1] http://aicrg.softwarefoundationfiji.org.

http://dx.doi.org/10.1016/j.asoc.2016.12.035
1568-4946/© 2016 Elsevier B.V. All rights reserved.

CNNs have been well suited for difficult problems that include recognition and detection [12] and can also be applied to large-scale video classification problems [16]. Howsoever, they have been mostly deployed for constrained and indoor vision applications that do not have problems of motion blur and noise which results from a moving camera. Therefore, the challenge is for them to be deployed for mobile devices. A cloud-based support system can be a solution to this problem of portability and computation power, however, good internet quality would be required for real-time implementation. Although mobile face detection and recognition has been getting popular [17], we gathered through the literature that there has not been much work done in the area of mobile face detection and recognition in unconstrained environments [18,19]. Mobile face detection and recognition consists of detection and recognition from a mobile source on stationary subjects and moving subjects which leads to input that contains motion blur and noise.

This paper presents a visual assistive system that features mobile face detection and recognition in an unconstrained environment from a mobile source using CNNs. The goal of the system is to effectively detect and recognise individuals who approach facing towards the person equipped with the system. Due to the shortage of related datasets, we present a dataset of videos captured from a mobile source that features motion blur and noise in an unconstrained environment from the mobile camera. This makes the application a very challenging aspect of face detection and recognition in unconstrained environments. The performance of the detection and recognition problems are evaluated using CNNs and cascade classifiers in different lighting conditions which include artificial light, daylight and moonlight.

The proposed approach contributes to a larger system designed to aid visually impaired persons through mobile face detection and recognition. We also provide a framework for implementation of the system with smartphones and wearable devices for video input and auditory notification from the system. This paper extends previous work that focused on face detection with CNNs [20] and mobile application framework [57].

The rest of the paper is organised as follows. We present the background and related work in Section 2 and the proposed mobile visual assistive system in Section 3. Section 4 describes the experimental design and also presents the experiment results. Section 5 gives a discussion and Section 6 concludes the paper with directions for future work.

## 2. Background and related work

### 2.1. Face detection and recognition

Face detection and recognition are the processes of verifying faces in a given environment via computer vision algorithms that usually involve machine learning [15,19]. Face recognition is performed in a wide range of conditions based on facial features, emerging technologies and learning algorithms [18,21]. Some of these methods use emerging technologies such as infra-red camera [22] and involve three-dimensional face recognition systems [23]. Some of the related methods for this paper are discussed as follows.

Ortiz et al. presented a recognition system that used a custom dataset of 800,000 web-scale face images from the *Facebook* social network to perform face recognition using a linearly approximated sparse representation-based classification algorithm (LASRC) [24]. LASRC achieved a speed-up by a factor of 100–250 when compared to the sparse representation classification (SRC) algorithm and also outperformed advanced algorithms on identification in uncontrolled web-scale setups where images are scaled to sizes suitable for use on the internet. Raghavendra et al. proposed a system that consisted of a face and speech module for performing verification [25]. The face module used a combination of principal component analysis (PCA) algorithms for feature extraction [26,27]. The speech module utilised text independent speaker verification using cepstral coefficients for feature extraction and Gaussian mixture model for the opinion generator. Furthermore, Pong et al. developed a face recognition system using multi-resolution feature fusion [28]. This system used information from face images at high and low resolutions to improve data stored in extracted features. A genetic algorithm was then used to combine the features into a single vector for recognition since the features were related to each other and showed promising results with related methods.

Face recognition using images presents problems such as variances in pose, lighting and scale [18,29]. Hence, video-based recognition reduces these problems since they can have multiple images (video frames) of a specific scene and have been previously used effectively for detection and recognition of facial expressions [30,31]. Stallkamp et al. presented a real-time video-based face identification system which recognised people entering through the door of a laboratory [29]. The subjects participating in experiments were asked not to cooperate with the system to create a challenging recognition scenario. The challenges included continuous variations in facial appearance due to illumination, pose, expression and occlusion. The classification of faces were done using an appearance-based recognition algorithm. The authors introduced three different measures to determine the contribution of each individual frame to the overall classification decision and reported a closed-set (subjects registered in the database) correct classification rate of 92.5% using $k$-nearest neighbours method and 91.8% when using Gaussian mixture models. Gorodnichy proposed another video-based framework for face recognition in videos [32]. This work made a clear differentiation between facial data obtained from images and those acquired from videos. They were considered to be two different modalities with one providing hard biometrics and the other providing softer biometrics. It was also shown that face images which had at least 12 pixels between the eyes could be recognised by computers and humans. A video database consisting of 11 people was introduced and the experiments showed a recognition rate of over 95%.

Video-based object detection and recognition systems typically have a fixed location for the camera which provides video frames as input. However, some applications such as autonomous driving systems and pedestrian detection systems also include a mobile input source [33,34]. These cases can increase the complexity of detection and recognition scenarios by introducing motion blur and having input frames lose track of the target since the source is also moving. Levinson et al. presented an autonomous driving system which focused on autonomous driving in real-world conditions [33]. The autonomous car was able to track and classify obstacles as cyclists, pedestrians and vehicles while driving. It was also capable of operating during the day or night and remained unaffected by weather conditions. Gavrila et al. proposed a multi-cue vision system for pedestrian detection and tracking from a moving vehicle [34]. The experiments in difficult urban traffic conditions showed a correct recognition percentage of 62–100% at the cost of 0.35 false classifications per minute.

Unconstrained face recognition considers a wide range of situations, where subjects are mobile and images and videos contain noise along with variations with different lighting conditions. Cox et al. presented an unconstrained face recognition approach that achieved high performance on the *Labeled Faces in the Wild (LFW)* [17] unconstrained face recognition challenge set [35]. The authors achieved this performance by combining machine learning methods with feature representations generated using brute-force search. Moreover, Ding et al. proposed a new scheme to extract

Multi-Directional Multi-Level Dual-Cross Patterns (MDML-DCPs) from face images for unconstrained face recognition that was robust to variations in illumination, pose and expression [36]. The scheme used the first derivative of the Gaussian operator to reduce the impact of illumination changes and compute the DCP (a face image descriptor based on the textural structure of human faces) features.

### 2.2. Convolutional neural networks

The basic idea of CNNs come from Hubel and Wisel's study of the visual cortex in 1962 [37]. Their work identified simple cells and complex cells which would later become related to filter bank layers and pooling layers used in CNNs. The first computer-based CNN simulation was done by Kunihiko Fukushima in 1980 [38]. This network which was known as *Neocognitron* was self-organising and recognised patterns based on the geometrical similarity of their shapes while remaining unaffected by small shape distortions. It improved on the performance of the previous *Cognitron* network [39] which depended on position changes by becoming invariant to them.

Matsugu et al. used CNNs for face detection and expression recognition in a system that was subject independent and resistant to differences in the positioning, rotation and scale of facial expressions [13]. The results showed a recognition accuracy of 97.6% for faces with a 'smiling expression' and the system was able to provide a distinction between 'smiling' and 'talking' faces. Ciresan et al. presented an object detection and recognition system that used graphics processing units (GPU) to implement CNNs and included feature extractors which learned in a supervised way [14]. The GPU based computation allowed the system to be fast while the parameterised nature of the network made it possible to adapt it to specific applications. The experiments were executed on benchmark databases [40–42] with very promising error rates. Several CNN training algorithms and architectures that further reduced the error rate on the MNIST database (the lowest achieved on the benchmark databases by the researchers) were developed later [43,44].

Recent works have seen development of new applications, approaches and systems using CNNs. For instance, Sun et al. presented a facial-point detection system where cascaded regression approach for facial-point detection was used with three levels of CNNs [15]. The first level used deep CNNs to make accurate predictions while the next two levels used shallow CNNs to refine the initial estimation of key points resulting in a high accuracy. Multiple CNNs at each level were fused to improve the accuracy and reliability of estimations. Experiments showed the new approach outperformed state-of-the-art methods on accuracy and reliability.

Apart from image processing, CNNs have also been applied to other areas such as video classification and speech recognition [16,45]. Karpathy et al. employed CNNs for large-scale video classification [16] that used a dataset of 1 million *YouTube* videos belonging to 487 classes of sports. The CNN architecture processed input at two spatial resolutions as a method of improving the runtime performance without affecting the accuracy. These resolutions consisted of a low-resolution context stream and a high-resolution fovea stream. The experiments reported an accuracy of 63.3% that improved previous accuracy of 43.9% and indicated that the learned features were generic and generalised other video classification tasks. Furthermore, Sainath et al. presented CNNs for speech recognition [45] in an architecture that made them more effective compared to deep neural networks (DNNs) for large vocabulary continuous speech recognition. The results showed that CNNs were able to achieve a 13–30% relative improvement over Gaussian mixture models (GMMs), and 4–12% relative improvement over DNNs.

### 2.3. Cascade classifiers

Cascade classifiers are ensemble learning methods that use a sequence of classifiers to make a decision. They are based on the concatenation of several classifiers where all the information collected from the output from a given classifier is used as additional information for the next classifier in the cascade [7]. The cascades in a cascade classifier refer to the resulting classifier created from many smaller classifiers (known as stages) which are applied sequentially to a specific region [46]. Each stage in the cascade makes a decision which determines the final output. They are suitable for mobile devices since they have low computational requirements [7].

Viola and Jones proposed the cascade classifier in 2001 as an object detection framework for the problem of face detection [7]. They introduced an image representation known as the integral image which allowed quick computation of features used by the detector The learning algorithm based on Ada-Boost [47] was used to obtain classifiers from important visual features. It can be used to create boosted classifiers which are classifiers at each stage of the cascade created from simple classifiers using a boosting technique [48]. A method for combining classifiers into a cascade structure was proposed in order to increase the detection speed.

Cascade classifiers have the advantage of computing features at an extremely fast speed and are also efficient at selecting features since false-positives are removed at an early stage reducing computation time required at the later stages [7]. They are also invariant to scale and location changes. Furthermore, the generic nature of cascade classifier allows them to be trained for detection of other types of objects not just faces [49]. However, cascade classifiers also have limitations such as being mostly effective on frontal face images. In addition, lighting conditions can affect the performance as the amount of light can determine whether features get detected. They are also sensitive to rotation changes of the object being detected [19].

## 3. Mobile visual assistive system

The proposed face detection and face recognition in an unconstrained environment is part of a mobile visual assistive system through a mobile application designed to assist visually impaired persons. We first describe the architecture of the system and their interaction and then provide their implementation details. We note that the major component is the *intelligent systems module* which can be implemented using either CNNs or cascade classifiers, depending on their performance from simulation in the next section. The overall system is described further in Fig. 1.

We employ cloud-based support system to feature computation in terms of training the selected components. The knowledge learnt is transferred into the mobile device through specific data transfer protocols in order to ensure security and reliability. In this way, the training is executed in the cloud computing infrastructure that helps in the elimination of the problems faced with power consumption and heat generation for high computational tasks in mobile devices [14].

The mobile visual assistive system has been designed with the goal to be implemented for low-end mobile devices that feature camera input. The system has two modes of operation, which are live and safe mode. In live mode, the user is either stationary or mobile with input from the live camera (*camera input*) coming from a wearable device. The wearable device consists of a camera typically mounted on a eye-glass frame which is connected to the mobile device that accesses the intelligent systems module on the cloud-based support system. When a frame is ready to be processed for either *detection* or *recognition*, the input image is

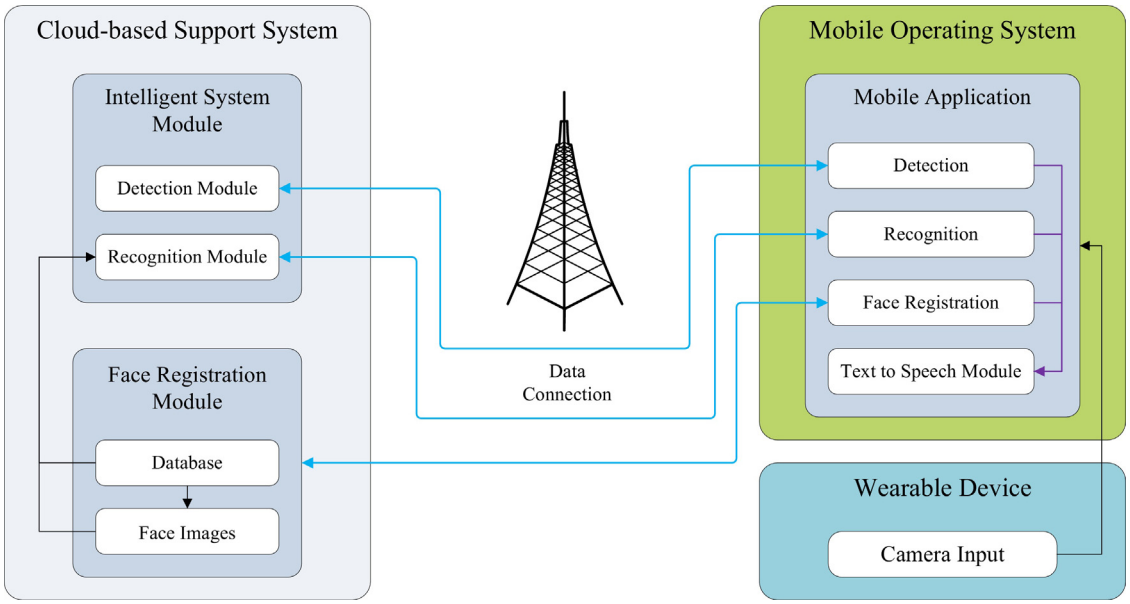



**Fig. 1.** The intelligent systems module of the mobile visual assistive system responds to detection and recognition requests made by the mobile application. When a new face needs to be enrolled for recognition, the *face registration module* saves the name of the person together with an image of the person's face. This recognition data can later be transferred to the mobile application enabling the device to perform recognition without a data connection.

sent to the *detection module* or *recognition module*. As detections and recognitions are made, the decision is regularly notified to the user through an earpiece using the *text to speech module*. In safe mode, the mobile application is cut-off from the cloud-based support system in a situation which could be due to the lack of Internet connectivity. However, it is capable of detection and recognition through the local modules as explained hereafter.

The intelligent system module consists of the *detection module* and *recognition module* which are used for training and performing either detection or recognition. The purpose of the *detection module* is to automatically detect faces while the *recognition module* identifies the detected faces with help from the *face registration module* that acts as a memory mechanism for the mobile visual assistive system. It communicates with the application running on the mobile device for identification. Hence, accounts of known faces are required which can be created by either uploading images from profiles, from integrated sources such as social networks or through the user's mobile device when they meet the person for the first time. Therefore, when the user meets a new person and needs to add them in the *face registration module*, a few snapshots are automatically taken by the mobile device which captures the face of the new person through *face registration* and sends them to the cloud-based support system. If the device successfully takes the snapshots it sends a notification to the user to add the name of the new person and then uploads the photo and name to the cloud-based system in order to be added to the *face registration module* as shown in Fig. 1.

The registration of people is motivated by the way humans remember faces when they recognise them. Therefore, further images can be added to the individuals already enrolled by the *face registration module* making recognition more robust to changes to the individual's face. These changes can be addition of sun-glasses, facial hair, etc. to the person's face. The *face registration module* is implemented using a database that stores the name of a person enrolled by that user as well as the filenames of images that belong to the enrolled person. It can also be used by friends and family members of the mobile assistive system user to enrol their face.

Knowledge of the recognition data on the cloud-based support system can be transferred to the mobile device through a *local face database* stored on the device. This *local face database* is created from a user defined amount of frequently recognised faces through the *face registration module* and allows the device to keep a recent version of the recognition data for use in safe mode. Although real-time implementations will encounter some privacy issues, we only provide a road-map for implementation and as such the privacy issue of enrolled users with policies can be focus of future research.

We further note that the technical implementations can vary for different types of cloud infrastructure, database software, wearable devices and mobile devices. We only provide general system architecture without focus on implementation details with the trend of cloud computing and mobile devices.

### 3.1. Convolutional neural networks

CNNs are essentially feedforward neural networks in which synaptic organisation between the neurons are inspired by the organization of the animal visual cortex [37]. CNNs are trainable multi-stage architectures (Fig. 2) which consist of multiple stages of image processing [12]. The input and output of each stage are sets of arrays known as feature maps. At the output, each feature map represents a particular feature extracted at all locations on the input. Each stage in the CNN uses an $n \times n$ kernel[2] of a specific size for processing the input image. In every stage, there are three layers which are mathematical functions; a filter bank layer, a non-linearity layer and a feature pooling layer. A typical CNN can be made up of one, two or three such 3-layer stages, followed by a classification module.

Each of the three layers in a stage of a CNN uses different methods to perform the respective computations. In the filter bank layer, a number of filters can be used to convolve the input image [12]. The definition of these filters depends on the problem. Low-level filters in the first convolution stage are normally edges and could be represented as diagonal horizontal and vertical lines. The non-linearity layer uses mathematical functions such as the hyperbolic tangent, sigmoid, and rectifier functions to make the learned features more

[2] A kernel is a small matrix in image processing used for performing various operations on an image such as blurring or edge detection.

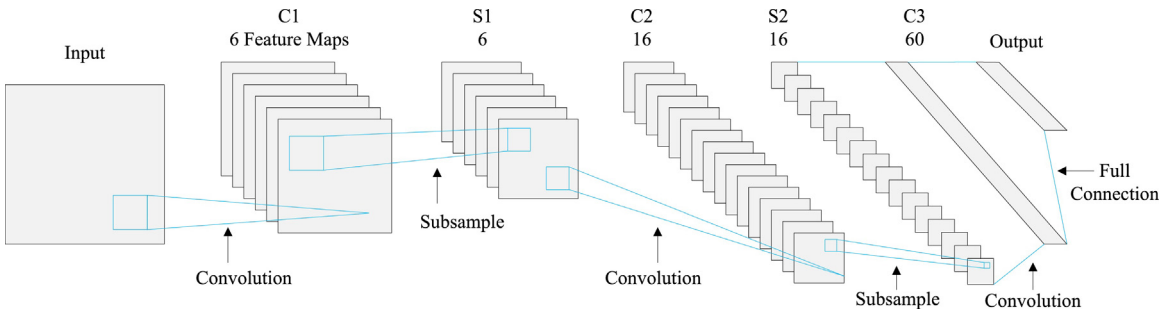



**Fig. 2.** Convolutional neural network architecture. Stages $C1$, $C2$ and $C3$ are convolution stages with a $n \times n$ kernel ($n = 5$) while stages $S1$ and $S2$ are sub-sampling stages with a $m \times m$ kernel ($m = 2$).

robust. The pooling layers typically use approaches such as average or maximum pooling. They compute the average or maximum of a particular feature over a specific region of the image in order to increase invariance.

In our CNN implementation for mobile face detection, we employed an architecture that incorporates three convolution stages and two sub-sampling stages (Fig. 2). Stages $C1$, $C2$ and $C3$ consist of a $n \times n$ kernel ($n = 5$) while stages $S1$ and $S2$ have a $m \times m$ kernel ($m = 2$). Each stage has an additive bias and a hyperbolic tangent (tanh) sigmoid function for the non-linearity layer. The convolution stages $C1$ and $C2$ also feature a subtractive normalisation layer with a $v \times v$ kernel ($v = 5$).

### 3.2. Simulation of the intelligent systems module

The general methodology follows a simulation study that views the videos taken in an unconstrained mobile environment as input for simulation of the user walking with the mobile device. We shall carry out the training of the intelligent system module that features CNNs and cascade classifier using an established face dataset; the cropped grey-scale version of the *Labeled Faces in the Wild* [17] database known as LFWcrop [50]. It should be noted that although the dataset is labelled,[3] the labels are not used in this paper for neither training nor experiments. Once trained, the intelligent module is tested with the videos that are custom made and features a wide range of conditions where the user is stationary or moving that leads to motion blur due to camera shakes. The following steps were taken to evaluate the performance of the intelligent systems module.

- Step 1: Record videos in a wide range of situations from moving and stationary positions.
- Step 2: Perform extraction of frames from the videos at the recorded resolution of $1088 \times 1920$. One in every two frames is saved during the extraction process. Using this ratio minimises the amount of frames lost during extraction since each scene in the video lasts for more than 1 min.
- Step 3: Resize the extracted frames to a resolution of $283 \times 500$ to reduce the computation time. Since the images are down-scaled from a higher resolution, there are very little negative effects to the quality of the images.
- Step 4: Train the respective detection and recognition methods (CNNs and cascade classifiers) using the LFWcrop dataset. This is an important part of the overall process since the training provides a major contribution to the accuracy of the detection and recognition methods.
- Step 5: Test detection and recognition methods from data (images) obtained from Step 3.
- Step 6: Measure performance by computing the detection rate for each video. First the correct and incorrect detections are manually verified by a human. Then the detection rate is calculated by taking the quotient of the correct number of detections and the overall number of detections.

In order to perform the face detection experiments, the frames from each video were first extracted. Note that videos are essentially a series of images given in time. Since the videos were recorded with a high resolution camera, each frame was then converted to a lower resolution image in order to decrease the computation time during detection and recognition. The Lanczos re-sampling algorithm was used for converting the images since it balances conversion speed and image quality [51].

For face recognition, the face images from the custom video data set were extracted by first performing face detection on all the videos. This is done to prepare a set of face images that can be used for recognition. Once the detection process was complete, the face images from billboards, posters, etc. were deleted since they do not belong to a person that is moving. The face images of each person were then grouped into folders with random names for the purposes of the recognition experiment. After the images were grouped, they were split into *training* and *testing* sets. These images are referred to as the *recognition data set* in the paper and are essentially derived from faces of people in the custom video data set. The *training* set is used for training the CNN and cascade classifier while the *testing* set is used to run experiments on the CNN and cascade classifier.

### 3.3. Mobile face detection and recognition dataset for unconstrained environments

Due to the novelty of the application problem and unavailability of specific data set for testing, a special data set was created using a smartphone camera to check the accuracy of the detection using CNNs.

The custom video data set [52] was created and used for the experiments as there are no existing data sets that fulfilled the requirements of our experiments. The data set consists of 11 videos recorded at a resolution of $1920 \times 1088$ at 29 frames-per-second (FPS) using a mobile phone camera placed inside a shirt pocket of the moving person. The videos are not recorded in controlled environments (which is the case for many other data sets), and hence provides real-world conditions where various challenges are present. Each video features a moving source (camera) and tests for face detection in difficult conditions which include different lighting conditions that includes motion blur, obstructions, rotations and scale changes as shown in Fig. 3.

[3] The label is the identity of the person in the face image.

(a) Artificial Lighting (b) Daylight

(c) Moonlight and Artificial Lighting (d) Artificial Lighting and Daylight

**Fig. 3.** Different lighting conditions of the dataset.

The different lighting conditions include artificial light, daylight and moonlight (Fig. 3a–c). A total of six videos were recorded for artificial lighting and daylight conditions with each condition having three video recordings. An additional of three videos were taken in both artificial lighting and daylight conditions. Furthermore, two videos were recorded in both artificial lighting and moonlight conditions. The videos with two lighting conditions (Fig 3d) include transitions from one condition to another. For instance, the video begins with daylight and then transits to artificial lighting as the camera moves from one location to another.

## 4. Simulation and results

We present simulation study of the proposed intelligent system module that features detection and recognition using CNNs. Cascade classifiers are used for further comparison of the results. We use the simulation study methodology and video dataset with wide range of conditions described in the previous section.

### 4.1. Design of experiments

The detection module was tested on frames from the custom video dataset [52] while the recognition programs were executed on the *testing* set of the *recognition data set* (refer to Section 3.2). The face detection and recognition modules were implemented using the EBLearn C++ library implementation of CNNs [53]. The cascade classifier was implemented using functions from the Open Source Computer Vision Library (OpenCV) [54]. These implementations use an Extensible Markup Language (XML) classifier that utilises Haar features for detection or recognition [7].

**Table 1**
The number of persons in the training set and testing set of the recognition data set.

| Recognition data set | |
|---|---|
| Training set | Testing set |
| 18 known | 61 (43 unknown + 18 known) |

A Linux system with a 2.2 Giga-hertz dual-core processor was used to run the experiments to depict the computation module in the cloud-based infrastructure. All the respective experiments for detection and recognition were conducted with 30 independent experimental runs in order to report the mean and standard deviation in the results.

The face detection training was done using face images from the LFWcrop [50] database and background images from the Caltech background image data set [55]. Face images from LFWcrop were overlaid on the background data set to create the final images used for training. After training was completed using the respective methods (CNN and cascade classifier), the detection experiments for the respective methods were conducted on frames extracted from the 11 videos of the custom video dataset. A total of 6111 frames were extracted from all of the videos. Each frame was then scaled down to a resolution of $283 \times 500$ by applying the Lanczos resampling algorithm [51]. The true and false positives (correct and incorrect detections) from the output files were manually verified after all experiments were finished.

Training for the recognition program was done using the *training* set of the *recognition data set* which consisted of images from a total of 18 different people (Table 1) and background images from the Caltech background image data set [55]. Face images from the *training* set of the *recognition data set* were also overlaid on the background data set to create the final training images.

Experiments for recognition were run on the *testing* set of the *recognition data set*. The *testing* set was comprised of images from 61 different individuals which included images of the 18 known people used in the *training* set. These known images belonged to the same 18 people and were basically different images of the same people. Hence, the total number of unknown individuals in the *testing* set were 43 (61–18). We note that although the number of unknown persons (43) is higher than the 18 used in the *testing* set, the total number of images for the 18 individuals is greater than the 43 unknown persons.

### 4.2. Results for detection

The results of the detection from the CNN is shown in Table 2. Each frame was processed on average in 1.61 s on a Linux system with a 2.2 GHz dual-core processor.

The results show that the performance of the detection module using CNN is poor in artificial lighting (Video 1) especially when there are only a few faces in all of the frames. Videos 2–3 show that average performance is achievable when conditions are more suitable. The best detection performance under artificial lighting conditions was achieved in Video 3. A timeline of this performance is shown in Fig. 5. Frames from Video 3 featured 6 unique faces with motion blur present in all frames resulting in a low number of detections. In normal daylight, however (Videos 4–5), the performance is good and depends on the difficulty of the video. High performance is attained in videos with relatively easy detection scenarios (Video

6), while the performance is lower in other videos. Video 6 provides the best detection performance in daylight (Fig. 4a) as shown in Fig. 6. This video consisted of faces from 7 different individuals with a high accuracy but was unable to detect faces in some frames due to motion blur.

The videos that have transitions between daylight and artificial lighting (Videos 8–9) also show good performance with only videos with difficult detection scenarios achieving poor performance (Video 7). This is due to a lack of light in a few frames where the transition between the lighting condition is made and constant rotation changes (the changing angle of the input frame) as each frame is processed. The best detection performance in two lighting conditions is achieved by Video 8. A timeline of the detection performance for Video 8 is shown in Fig. 7. Moreover, Video 8 included 13 unique faces and lower motion blur resulting in a higher detection rate. Furthermore, the videos that include moonlight and artificial lighting (Videos 10–11) leads to poor detection. This is caused by lack of lighting on the faces and motion blur as shown in Fig. 4b.

The results from the cascade classifier is shown in Table 3. The average computation time for each frame was 265 ms on a Linux system with a 2.2 GHz dual-core processor. We find that the performance of the cascade classifier is poor in artificial lighting conditions (Videos 1–2) where the best detection rate is only reported in bright light (Video 3). We observe poor performance in daylight conditions (Videos 4–5), only the stable videos (Video 6) showed capable performance. The cascade classifier is also poor in transitional lighting videos (Videos 7 and 9) where the lighting gradually changes from light to dark and vice-versa. We find that good performance is only achieved when the faces are nearer to the camera (Video 8). Furthermore, we observe very poor performance in moonlight and artificial lighting conditions (Videos 10–11). The minimal lighting on faces and motion blur leads to a high number of false-positives. It can be gathered that the cascade classifier performance is high for simple detection tasks and for stable videos which include proper lighting.

### 4.3. Results for recognition module

The results of the recognition experiments based on CNNs are shown in Table 4. Each recognition image for a person was processed on average in 1.14 s on a Linux system with a 2.2 GHz dual-core processor. We observe 100% face recognition for some of the cases (Persons 1, 3–5 and 7–8). This ideal performance is caused by similarities between the training and test images such as lighting conditions and rotation of faces (angle at which the face is shown).

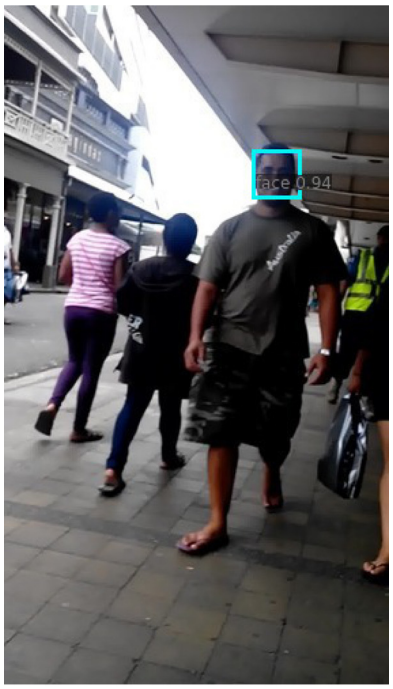
Lower performance is achieved by the recognition program from the images of Person 6, 12–14 and 18. The lower performance can be attributed to the different faces having some resemblance. This is due to the distance from the camera and the angle at which the image is captured depending on the pose of the video capture source and the subject. The CNN fails to correctly identify faces from images of Person 2, 9–11 and 15–17. It is apparent that poor face resemblance and lighting conditions lead to poor recognition rate.

The results from the cascade classifier are shown in Table 5. During the experiments, the average processing time for each image was 568.72 ms.

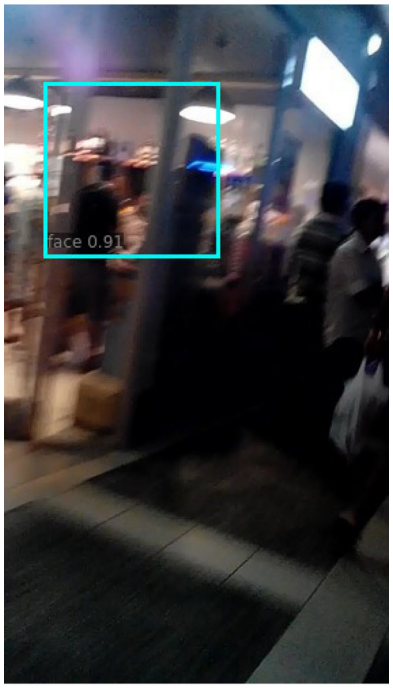
The recognition rate of the cascade classifier program is also 100% for images of Person 2, 5, 7, 9–10, 13 and 16–18. The perfect performance is due to similarities in lighting conditions and rotation of faces (angle at which the face is shown).

We observe that better performance is shown by experiments done on images of Person 1, 3, 6, 8, 11 and 14–15. The incorrect identifications are due to the confusion of a person's face with others from the similarity of facial features and the presence of motion blur and noise. The algorithm is unable to distinguish between them – a task that humans can somewhat easily accomplish given well-lit conditions. The performance on Persons 4 and 12 shows an intensified case of this situation since the performance is lower compared to the performance of the other cases.

The cascade classifier shows real-world performance from experiments on the majority of the images when the ideal performance is disregarded as an accuracy of 100% cannot be obtained in practical scenarios.




(a) Correct




(b) Incorrect

**Fig. 4.** Examples of correct and incorrect face detection.

**Table 2**
CNN detection results from the previous work.

| Video | Condition | Frames | Detection | Correct | Incorrect | Average detection rate (%) | Error (%) |
|---|---|---|---|---|---|---|---|
| 1 | | 296 | 15 | 2 | 13 | 13.33 | 1.49 |
| 2 | Artificial light | 518 | 14 | 9 | 5 | 64.29 | 1.49 |
| 3 | | 359 | 9 | 6 | 3 | 66.67 | 1.49 |
| 4 | | 229 | 46 | 27 | 19 | 58.70 | 1.49 |
| 5 | Daylight | 122 | 10 | 6 | 4 | 60.00 | 1.49 |
| 6 | | 224 | 56 | 53 | 3 | 94.64 | 1.49 |
| 7 | | 862 | 27 | 12 | 15 | 44.44 | 0.78 |
| 8 | Artificial light &daylight | 774 | 191 | 172 | 19 | 90.05 | 0.78 |
| 9 | | 1206 | 124 | 105 | 19 | 84.68 | 1.49 |
| 10 | Moonlight &artificial light | 1047 | 16 | 0 | 16 | 0.00 | 0 |
| 11 | | 474 | 16 | 2 | 14 | 12.50 | 1.49 |

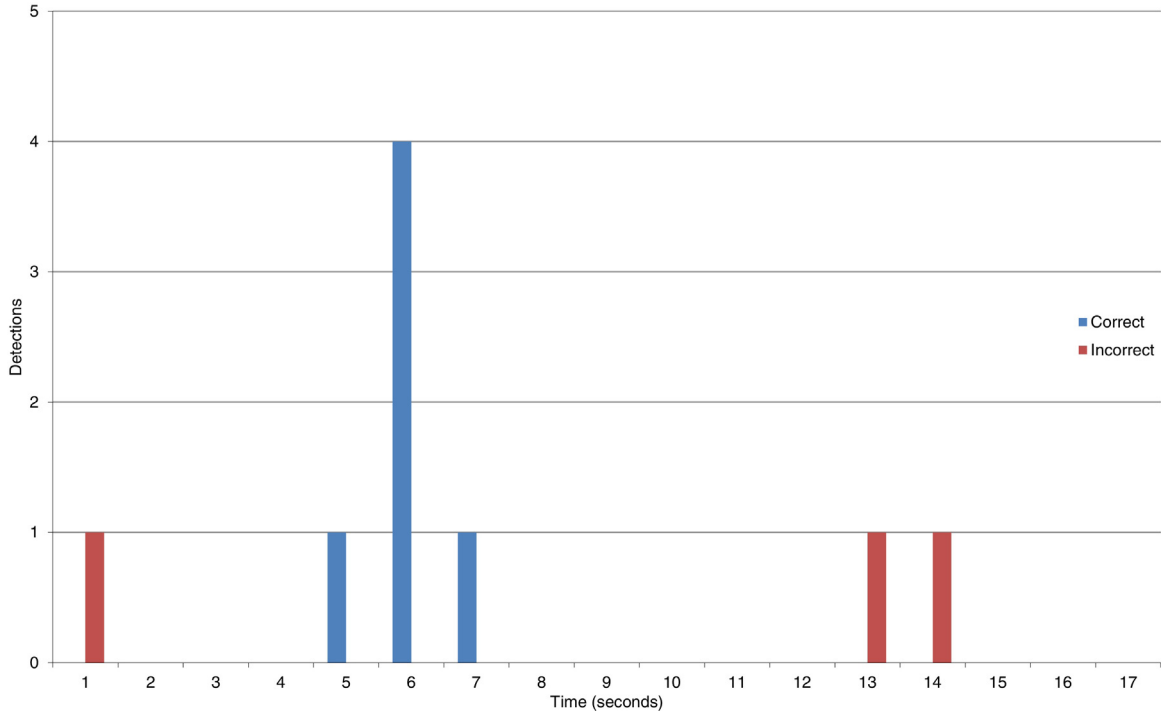



**Fig. 5.** Timeline of detections for Video 3. Each second represents 18 frames.

The comparison of performance of CNNs and cascade classifier for the face recognition modules is shown in Fig. 8 while the detection module is shown in Fig. 9. We observe that the face detection is better using CNNs with an accuracy of 94.64% for almost all the cases, except for Video 1 and Video 10 where motion blur and poor lighting conditions are better handled by the cascade classifier.

In general, face recognition has been better when cascade classifiers are used. When training for face detection, the custom video data set is used in order to detect as many faces as possible while

**Table 3**
Cascade classifier detection results from previous work.

| Video | Condition | Frames | Detection | Correct | Incorrect | Average detection rate (%) | Error (%) |
|---|---|---|---|---|---|---|---|
| 1 | | 296 | 69 | 23 | 46 | 33.33 | 1.65 |
| 2 | Artificial light | 518 | 78 | 18 | 60 | 23.08 | 1.34 |
| 3 | | 359 | 84 | 45 | 39 | 53.57 | 1.48 |
| 4 | | 229 | 146 | 59 | 87 | 40.41 | 1.28 |
| 5 | Daylight | 122 | 83 | 40 | 43 | 48.19 | 1.46 |
| 6 | | 224 | 82 | 48 | 34 | 58.54 | 1.34 |
| 7 | | 862 | 62 | 17 | 45 | 27.42 | 0.80 |
| 8 | Artificial light &daylight | 774 | 214 | 158 | 56 | 73.83 | 0.76 |
| 9 | | 1206 | 428 | 141 | 287 | 32.94 | 1.49 |
| 10 | Moonlight &artificial light | 1047 | 191 | 9 | 182 | 4.71 | 2.06 |
| 11 | | 474 | 119 | 11 | 108 | 9.24 | 2.1 |

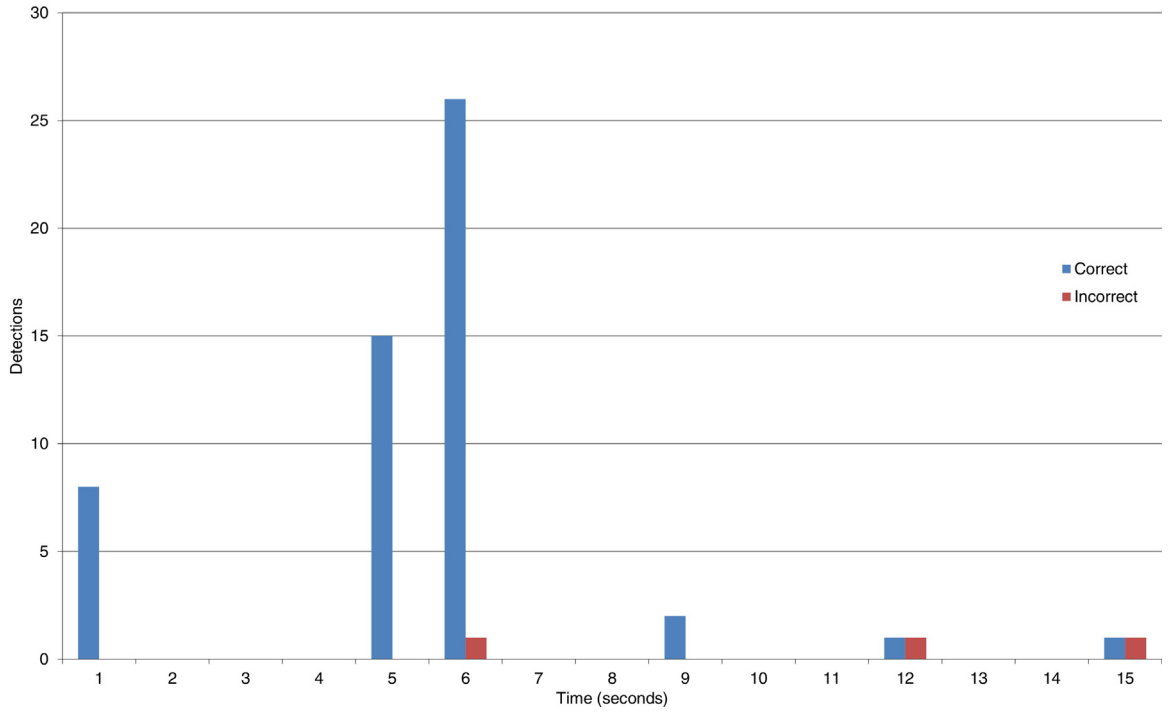



**Fig. 6.** Detection timeline of Video 6. Each second represents 15 frames except the last second where 14 frames are processed.

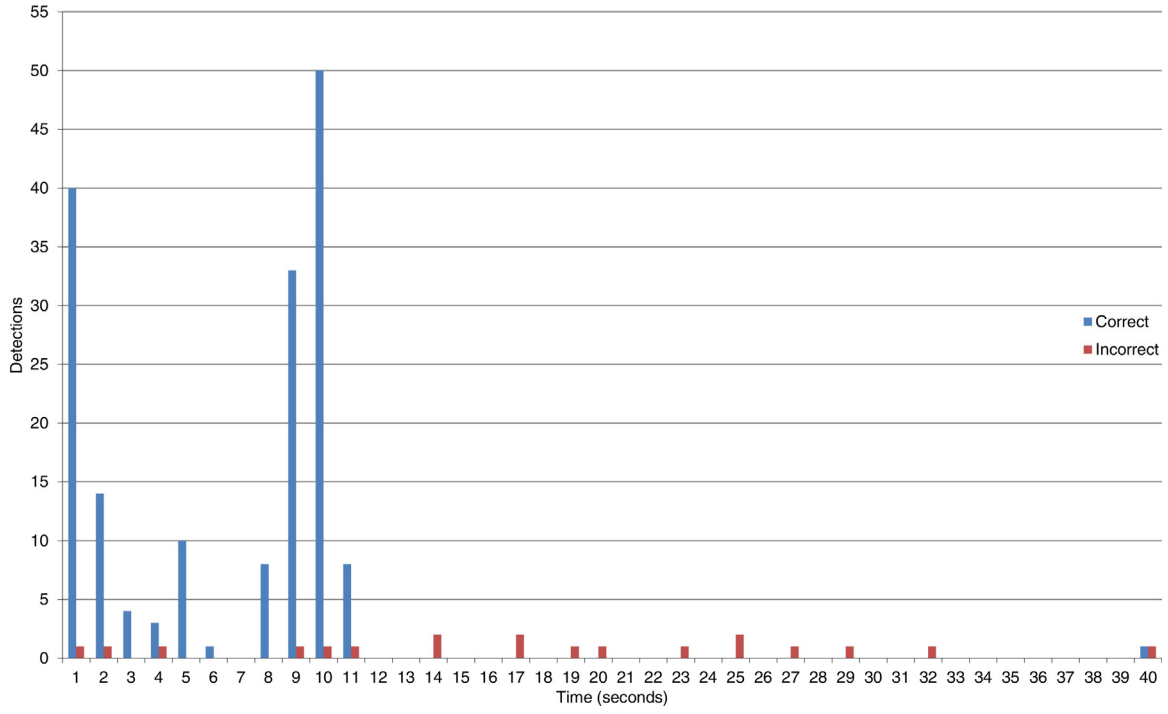



**Fig. 7.** Detection timeline of Video 8. Every second represents 16 frames.

in the recognition stage, a smaller data set (recognition data set) is used where the training is focused on specific faces.

## 5. Discussion

In the experiments, CNNs were compared to cascade classifiers. The performance of the cascade classifier recognition system makes it suitable for deployment on mobile devices since it performs computation in a short amount of time and therefore uses less energy. This makes cascade classifiers suitable for use in scenarios where internet connectivity is not available due to weak mobile network signals, weather conditions, etc. In this case, the *mobile application* can use cascade classifier-based detection and recognition on the device as a backup solution.

However, CNNs can provide better accuracy to users of the *mobile application* through the *intelligent systems module* (Fig. 1) if an internet connection is available. Furthermore, CNNs will not face performance degradation problems that cascade classifiers may

**Table 4**
Convolutional neural network recognition results.

| Person | Recognitions | Correct | Incorrect | Average accuracy (%) | Error (%) |
|---|---|---|---|---|---|
| 1 | 3 | 3 | 0 | 100 | 0 |
| 2 | 1 | 0 | 1 | 0 | – |
| 3 | 4 | 4 | 0 | 100 | 0 |
| 4 | 2 | 2 | 0 | 100 | 0 |
| 5 | 1 | 1 | 0 | 100 | 0 |
| 6 | 2 | 1 | 1 | 50 | 0.6616 |
| 7 | 2 | 2 | 0 | 100 | 0 |
| 8 | 1 | 1 | 0 | 100 | 0 |
| 9 | 0 | 0 | 0 | 0 | – |
| 10 | 0 | 0 | 0 | 0 | – |
| 11 | 1 | 0 | 1 | 0 | – |
| 12 | 2 | 1 | 1 | 50 | 0.7246 |
| 13 | 3 | 1 | 2 | 33.33 | 1.5751 |
| 14 | 2 | 1 | 1 | 50 | 0.7876 |
| 15 | 1 | 0 | 1 | 0 | – |
| 16 | 0 | 0 | 0 | 0 | – |
| 17 | 0 | 0 | 0 | 0 | – |
| 18 | 5 | 4 | 1 | 80 | 0.3938 |

have due to the increasing the number of images stored on the device as more people are enrolled for recognition. Each new image enrolled for recognition will require the recognition data cache to be rebuilt thus increasing the time needed by the *mobile application*. In addition, CNNs have proven to achieve state-of-the-art results in literature [15].

The *text to speech module* of the *mobile application* can be implemented through *Google TalkBack* [56], an application designed for assisting blind and vision-impaired users in interaction with their devices. It provides spoken, audible and vibration feedback to the user.

Wearable devices such as smartwatches or optical head-mounted displays (e.g. *Google Glass*, *Microsoft HoloLens*, etc.) can be used to provide the *camera input* to the *mobile application* enhancing the practicality and convenience of the mobile assistive system.

For the experiments, the frames extracted from videos in the custom video data set were directly used as input for detection and recognition. A number of frames included challenges such as different lighting conditions, scale, motion blur and obstructions. This unconstrained environment led to the poor performance of the detection (Table 2) and recognition (Table 4) algorithms. Pre-processing the input frames to reduce motion blur can be one of the approaches for improving performance. The recognition rate can be further improved by linking social networks to the face registration module and effectively increasing the number of images available for training the CNN.

On the other hand, the current performance in terms of speed is promising as the average computation time for a single image is 1.61 s for detection and 1.14 s for recognition on a laptop with a 2.2 sGHz dual-core processor. In a real-world implementation,

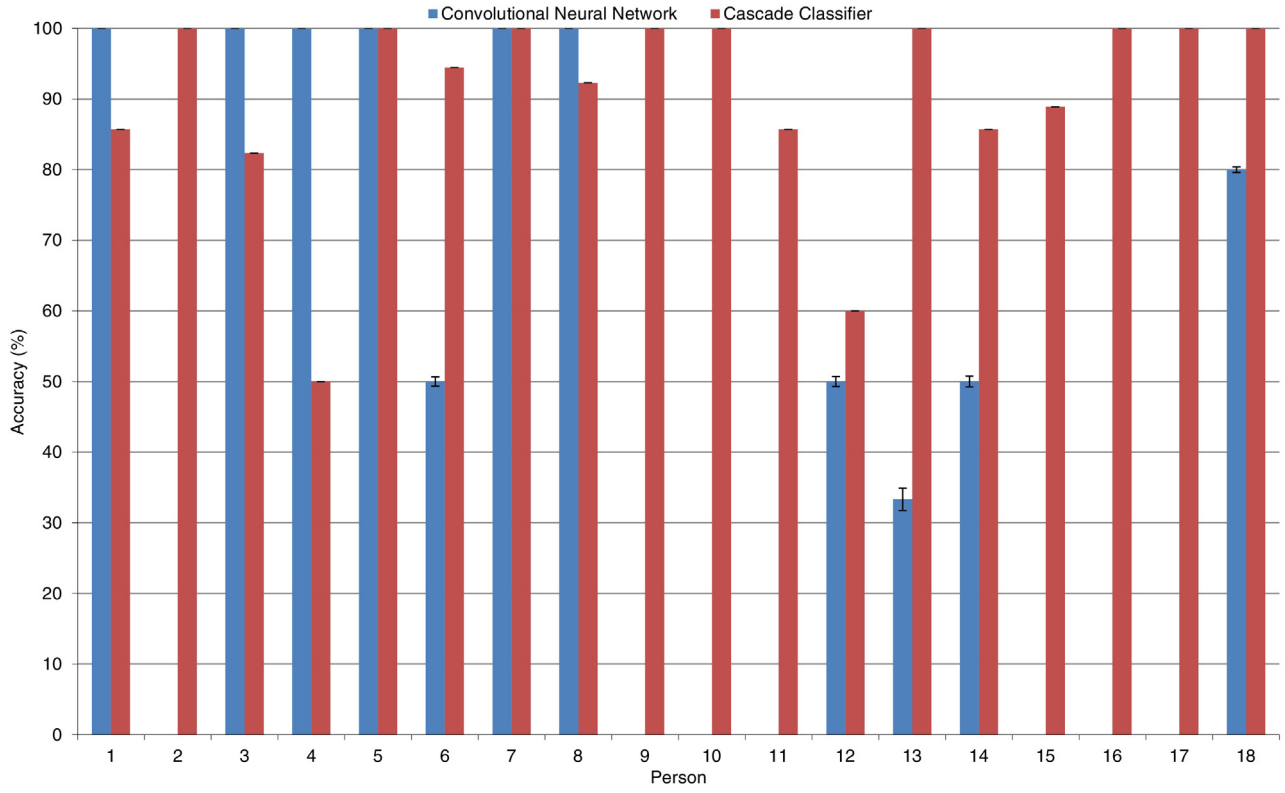



**Fig. 8.** Performance comparison for face recognition.

**Table 5**
Cascade classifier recognition results.

| Person | Recognitions | Correct | Incorrect | Average accuracy (%) | Error (%) |
|---|---|---|---|---|---|
| 1 | 14 | 12 | 2 | 85.71 | 1.49E−04 |
| 2 | 9 | 9 | 0 | 100 | 0 |
| 3 | 17 | 14 | 3 | 82.35 | 1.01E−04 |
| 4 | 10 | 5 | 5 | 50 | 6.53E−04 |
| 5 | 11 | 11 | 0 | 100 | 0 |
| 6 | 18 | 17 | 1 | 94.44 | 1.01E−04 |
| 7 | 9 | 9 | 0 | 100 | 0 |
| 8 | 13 | 12 | 1 | 92.31 | 1.04E−04 |
| 9 | 14 | 14 | 0 | 100 | 0 |
| 10 | 10 | 10 | 0 | 100 | 0 |
| 11 | 7 | 6 | 1 | 85.71 | 1.49E−04 |
| 12 | 10 | 6 | 4 | 60 | 6.53E−04 |
| 13 | 21 | 21 | 0 | 100 | 0 |
| 14 | 14 | 12 | 2 | 85.71 | 1.49E−04 |
| 15 | 9 | 8 | 1 | 88.89 | 1.04E−04 |
| 16 | 4 | 4 | 0 | 100 | 0 |
| 17 | 7 | 7 | 0 | 100 | 0 |
| 18 | 15 | 15 | 0 | 100 | 0 |

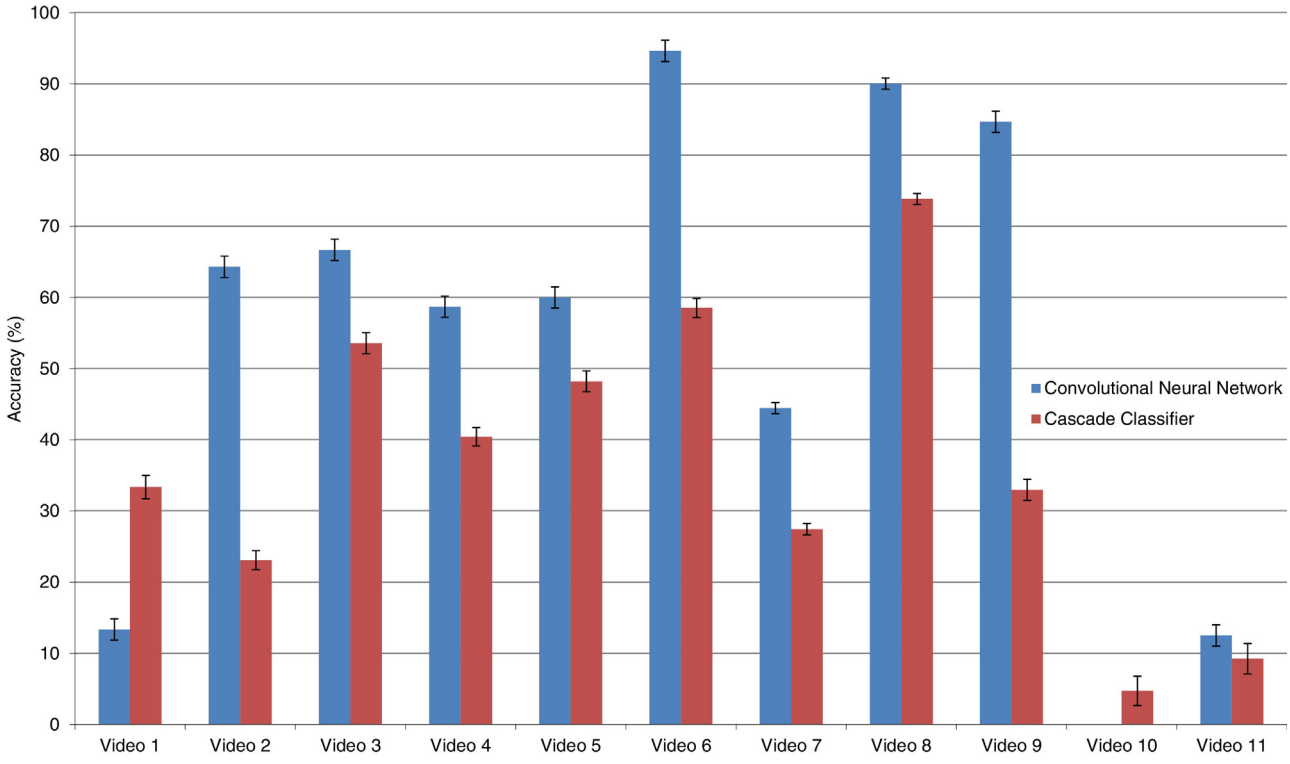



Figure 9: Performance for comparison face detection.

**Fig. 9.** Performance for comparison face detection.

the computation time can be further reduced through the use cloud-based computing infrastructure that implements parallel processing and graphic processing unit (GPU) based implementation for CNNs.

## 6. Conclusions and future work

This paper presented a visual assistive system that features mobile face detection and recognition in an unconstrained environment from mobile source using CNNs. The system's *intelligent systems module* included a *detection module* for face detection and *recognition module* for face recognition. The performance of the modules were evaluated using CNNs and cascade classifiers in different lighting conditions which included artificial light, daylight and moonlight. A dataset of videos captured from a mobile source that features motion blur and noise from camera shakes was created for the performance evaluation.

The highest performance achieved by the *detection module* using CNNs was 94.64% in daylight conditions compared to lower performance in other lighting conditions. Likewise, a maximum performance of 80% was obtained by the *recognition module* when using CNNs with poor performance coming from face images which had poor lighting. This implies that the detection and recognition algorithms need to be revised in future work to make them more robust to changes in lighting conditions and motion blur. In addition, the algorithms can be applied to other areas of object detection such as landmark recognition. This can be used to assist visually

impaired users in navigation by recognising landmarks (bus stops, grocery stores, etc.) specified by the user.

Furthermore, future work could focus on the use of transfer learning methodologies for CNNs that makes use of different datasets which includes frontal faces and those in unconstrained environments. The face detection and recognition systems modules could be further extended for object recognition. Moreover, facial expression recognition in an unconstrained mobile environment can also be the focus of future work. The emerging technologies such as three-dimensional face recognition with more than one camera input can be explored for real-time mobile applications that can guide the visually impaired. The use of infra-red camera can be helpful for night vision tasks and related datasets would be required for innovative research in this field

## References


[1] J. Zhou, D. Gao, D. Zhang, Moving vehicle detection for automatic traffic monitoring, IEEE Trans. Veh. Technol. 56 (1) (2007) 51–59.
[2] C. Lu, N. Adluru, H. Ling, G. Zhu, L.J. Latecki, Contour based object detection using part bundles, Comput. Vis. Image Underst. 114 (7) (2010) 827–834.
[3] A. Ess, B. Leibe, K. Schindler, L. Van Gool, Moving obstacle detection in highly dynamic scenes, in: ICRA'09, IEEE International Conference on Robotics and Automation, 2009, 2009, pp. 56–63.
[4] Q. Xie, J. Kim, Y. Wang, D. Shin, N. Chang, M. Pedram, Dynamic thermal management in mobile devices considering the thermal coupling between battery and application processor, in: Proceedings of the International Conference on Computer-Aided Design, ICCAD'13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 242–247.
[5] N.S. Latman, E. Herb, A field study of the accuracy and reliability of a biometric iris recognition system, Sci. Justice 53 (2) (2013) 98–102.
[6] M. Moreno, S. Shahrabadi, J. José, J. du Buf, J. Rodrigues, Realtime local navigation for the blind: detection of lateral doors and sound interface, Proc. Comput. Sci. 14 (0) (2012) 74–82, Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2012).
[7] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, CVPR 2001, vol. 1, IEEE, 2001, pp. 511–518.
[8] E. Guresen, G. Kayakutlu, Definition of artificial neural networks with comparison to other networks, Proc. Comput. Sci. 3 (0) (2011) 426–433, World Conference on Information Technology.
[9] J.M.H. du Buf, J. Barroso, J.M.F. Rodrigues, H. Paredes, M. Farrajota, H. Fernandes, J. José, V. Teixeira, M. Saleiro, The smartvision navigation prototype for blind users, JDCTA Int. J. Dig. Content Technol. Appl. (2011) 361.
[10] S. Willis, S. Helal, RFID information grid for blind navigation and wayfinding, in: ISWC, vol. 5, 2005, pp. 34–37.
[11] J.-S. Kang, Mobile iris recognition systems: an emerging biometric technology, Proc. Comput. Sci. 1 (1) (2010) 475–484, iCCS 2010.
[12] Y. LeCun, K. Kavukvuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proc. International Symposium on Circuits and Systems (ISCAS'10), IEEE, 2010.
[13] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network, Neural Netw. 16 (5) (2003) 555–559.
[14] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence – vol. 2, IJCAI'11, AAAI Press, 2011, pp. 1237–1242.
[15] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR'13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 3476–3483.
[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1725–1732.
[17] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.
[18] R. Jafri, H.R. Arabnia, A survey of face recognition techniques, J. Inf. Process. Syst. 5 (2) (2009) 41–68.
[19] C. Zhang, Z. Zhang, A survey of recent advances in face detection, Tech. Rep. MSR-TR-2010-66, June 2010 http://research.microsoft.com/apps/pubs/default.aspx?id=132077.
[20] S. Chaudhry, R. Chandra, Unconstrained Face Detection from a Mobile Source Using Convolutional Neural Networks, Springer International Publishing, cham, 2016, pp. 567–576.
[21] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: Computer Vision – ECCV 2004, Lecture Notes in Computer Science, vol. 3021, Springer Berlin Heidelberg, Germany, 2004, pp. 469–481.
[22] X. Chen, P.J. Flynn, K.W. Bowyer, IR and visible light face recognition, Comput. Vis. Image Underst. 99 (3) (2005) 332–358 http://www.sciencedirect.com/science/article/pii/S1077314205000226.
[23] Y. Lei, M. Bennamoun, M. Hayat, Y. Guo, An efficient 3d face recognition approach using local geometrical signatures, Pattern Recognit. 47 (2) (2014) 509–524.
[24] E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Underst. 118 (0) (2014) 153–170.
[25] R. Raghavendra, A. Rao, G.H. Kumar, Multimodal person verification system using face and speech, Proc. Comput. Sci. 2 (0) (2010) 181–187, Proceedings of the International Conference and Exhibition on Biometrics Technology.
[26] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev.: Comput. Stat. 2 (4) (2010) 433–459.
[27] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.
[28] K.-H. Pong, K.-M. Lam, Multi-resolution feature fusion for face recognition, Pattern Recognit. 47 (2) (2014) 556–567.
[29] J. Stallkamp, H. Ekenel, R. Stiefelhagen, Video-based face recognition on real-world data, in: IEEE 11th International Conference on Computer Vision, 2007, ICCV 2007, 2007, pp. 1–8.
[30] M. Hayat, M. Bennamoun, An automatic framework for textured 3d video-based facial expression recognition, IEEE Trans. Affect. Comput. 5 (3) (2014) 301–313.
[31] M. Abd El Meguid, M. Levine, Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers, IEEE Trans. Affect. Comput. 5 (2) (2014) 141–154.
[32] D. Gorodnichy, Video-based framework for face recognition in video, in: The 2nd Canadian Conference on Computer and Robot Vision: Proceedings, 2005, 2005, pp. 330–338.
[33] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, S. Thrun, Towards fully autonomous driving: systems and algorithms, in: Intelligent Vehicles Symposium (IV), 2011, IEEE, 2011, pp. 163–168.
[34] D. Gavrila, S. Munder, Multi-cue pedestrian detection and tracking from a moving vehicle, Int. J. Comput. Vis. 73 (1) (2007) 41–59.
[35] D. Cox, N. Pinto, Beyond simple features: a large-scale feature search approach to unconstrained face recognition, in: 2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE, 2011, pp. 8–15.
[36] C. Ding, J. Choi, D. Tao, L.S. Davis, Multi-directional multi-level dual-cross patterns for robust face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 518–531.
[37] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. 160 (1) (1962) 106.
[38] K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36 (4) (1980) 193–202.
[39] K. Fukushima, Cognitron: a self-organizing multilayered neural network, Biol. Cybern. 20 (3–4) (1975) 121–136.
[40] F.J. Huang, Y. LeCun, The NORB Dataset, 2015 http://www.cs.nyu.edu/ylclab/data/norb-v1.0/.
[41] A. Krizhevsky, V. Nair, G. Hinton, The CIFAR-10 Dataset, 2015 http://www.cs.toronto.edu/kriz/cifar.html.
[42] Y. LeCun, C. Cortes, C.J. Burges, The MNIST Database of Handwritten Digits, 2014 http://yann.lecun.com/exdb/mnist/.
[43] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 1135–1139.
[44] C. Wu, W. Fan, Y. He, J. Sun, S. Naoi, Handwritten character recognition by alternately trained relaxation convolutional neural network, in: 2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2014, pp. 291–296.
[45] T. Sainath, A.-R. Mohamed, B. Kingsbury, B. Ramabhadran, Deep convolutional neural networks for LVCSR, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, 2013, pp. 8614–8618.
[46] Z. Sun, Y. Wang, T. Tan, J. Cui, Improving iris recognition accuracy via cascaded classifiers, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 35 (3) (2005) 435–441.
[47] Y. Freund, R.E. Schapire, A short introduction to boosting, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1999, pp. 1401–1406.
[48] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.
[49] C. Premebida, O. Ludwig, M. Silva, U. Nunes, A cascade classifier applied in pedestrian detection using laser and image-based features, in: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2010, pp. 1153–1159.
[50] C. Sanderson, LFWcrop Face Dataset, 2014 http://conradsanderson.id.au/lfwcrop/.
[51] C.E. Duchon, Lanczos filtering in one and two dimensions, J. Appl. Meteorol. 18 (8) (1979) 1016–1022.

[52] AICRG Moving Object Video Dataset, 2015 http://aicrg.softwarefoundationfiji.org/open-source-software/aicrg-moving-object-video-dataset.
[53] Computational and Biological Learning Laboratory, New York University, Eblearn Home, 2015 http://eblearn.sourceforge.net/doku.html.
[54] OpenCV Developers Team, OpenCV | OpenCV, 2016 http://opencv.org/ (accessed: 30.08.16).
[55] M. Weber, Background Image Dataset, 2014 http://www.vision.caltech.edu/archive.html.
[56] Google Inc., Google TalkBack, 2016 https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=en (accessed: 30.08.16).
[57] S. Chaudhry, R. Chandra, Design of a Mobile Face Recognition System for Visually Impaired Persons, 2015, CoRR abs/1502.00756 https://arxiv.org/abs/1502.00756.