Dehydration Scan: An Artificial Intelligence Assisted Smartphone-Based System for Early Detection of Dehydration

Priyeta Saha(✉), Syed Muhammad Ibne Zulfiker, Tanzima Hashem, and Khandker Aftarul Islam

Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh
{1605094,1605110,1605063}@ugrad.cse.buet.ac.bd,
tanzimahashem@cse.buet.ac.bd

Abstract. Dehydration occurs due to fluid loss from the human body, affects regular body functions, and causes health complications. Physical exercises, poor fluid intake, and diseases like fever and diarrhea may result in dehydration. Current clinical and laboratory-based dehydration detection techniques are expensive, time-consuming, and require people to visit medical facilities, which often do not exist in destitute areas. Though recent research has focused on monitoring physiological parameters (e.g., heart rate, stress, and oxygen) and detecting diseases using smartphones, the area of dehydration detection has not been sufficiently addressed. We present a smartphone-based early dehydration detection system using artificial intelligence, which is ubiquitous, quick, and does not require any additional cost or expertise to operate. We develop a siamese network-based deep learning model to detect the changes in the facial landmarks that appear from dehydration and are not detectable with the naked eyes of general people. Our model provides an overall accuracy of 76.1% and is lightweight enough to run on a smartphone processor. By integrating it in the background, we develop a smartphone app, "Dehydration Scan" that simply captures facial images of individuals and detects their hydration status. Knowing early about dehydration allows people to take oral rehydration solutions and avoid severe dehydration.

Keywords: Dehydration detection · Mobile image analysis · Deep learning

1 Introduction

Dehydration occurs when the level of fluid in the human body falls below a certain threshold. Dehydration is a common effect of many diseases like fever,

P. Saha and S. M. I. Zulfiker—Both authors contributed equally to this research.

A. Tsanas and A. Triantafyllidis (Eds.): PH 2022, LNICST 488, pp. 289–312, 2023.
https://doi.org/10.1007/978-3-031-34586-9_20

diarrhoea and cholera, making it a significant cause of death worldwide. Lack of clean drinking water and proper sanitation causes people, especially in developing countries, to get infected with waterborne diseases like diarrhoea and cholera. Most people do not realize that they are dehydrated until it is too late. By then, hospitalization becomes a requirement. However, most of these people do not have economic solvency, nor do the hospitals have adequate human resources to provide proper healthcare in impoverished areas. Thus an increase in the death count becomes inevitable. Although these circumstances are more prevalent in lower-income countries, they exist in the developed world as well. People can also undergo mild to moderate levels of dehydration due to poor fluid intake or physical exercise. Studies [2,12] show that mild dehydration has an impact on alertness, concentration, mood, and cognitive abilities. Considering all the direct and indirect effects of dehydration on the human body and mind, a readily available diagnostic solution for the masses is a necessity at this point.

Compared to other mobile healthcare domains [11,21,22,32], researchers have not explored the topic of dehydration detection as much. Current dehydration detection techniques are primarily based on laboratory tests (blood or urine sample analysis) and clinical assessments undertaken by health professionals. Laboratory tests are expensive, time-consuming, and require specialized equipment. World Health Organization's IMCI algorithm [24] and DHAKA score [17] provide guidelines to health workers to detect dehydration based on clinical signs (e.g., sunken eye, skin turgor) and symptoms (e.g., thirst, pulse). However, health workers often do not have sufficient skills to follow the guidelines accordingly [1]. To reduce the need for skilled health workers or expensive medical equipment, researchers have developed dehydration detection techniques through kinetic analysis of hemoglobin concentration [33], exploiting photoplethysmographic signals [25] and measuring skin conductance [18]. However, they all require specialized hardware, making them expensive and inaccessible to the masses.

Dehydration causes very subtle changes to our facial landmarks, especially in regions like eyes and lips (e.g., reduced skin turgor or elasticity, tired and dry eyes, dry lips) [3,7,27]. These changes are not often perceivable by human eyes, especially when a person is in the early stages of dehydration. Therefore, it often goes unnoticed despite being a prevalent condition. We utilize artificial intelligence to work around the limitations of human vision in this case. On the other hand, with the widespread use of smartphones containing high-resolution cameras, mobile image analysis has emerged as a convenient solution for medical diagnosis [19,21,22]. We develop a smartphone-based early dehydration detection system using artificial intelligence that overcomes the limitations of the existing solutions:

- *Accessible* - Medical facilities or specialized hardware are not equally available worldwide, whereas smartphones are accessible almost everywhere.
- *Quick* - Our smartphone-based application provides instantaneous results, whereas diagnostic tests at hospitals usually take one or more days to deliver reports.

- *Automated* - Our smartphone-based detection tool uses artificial intelligence to detect dehydration and does not depend on the skill of the health workers for observing clinical signs.
- *Easy to Use* - The built-in high-resolution cameras in modern-day smartphones make the detection straightforward for users.
- *No Cost* - Using our smartphone-based detection system, patients can make initial or even intermediate-level assessments without taking expensive and invasive diagnostic tests at hospitals or buying specialized hardware.
- *Early Detection* - Early detection using smartphones can prevent further deterioration of a patient's condition. Mild dehydration detection using our application can save users from reaching a state where they need hospitalization.

We develop a deep learning model to detect the changes in facial landmarks when an individual becomes dehydrated. A traditional classifier [6,14,26] based on facial images would not perform well since the changes caused by mild dehydration are minute and can vary from person to person. For example, the dehydrated face of an individual with dry skin is not the same as that of an individual with oily skin. Similarly, the dehydrated face of an older adult is different from that of a young person. To overcome this challenge, we adopt the siamese neural network [31], proposed for a context similar to ours. Our model takes two images as input: one hydrated facial image as the reference and the other facial image representing the current state (hydrated or dehydrated) of the individual. The model predicts the class for the current facial image as hydrated or dehydrated. In Sect. 5, we elaborate on how we adopt the siamese network-based contrast learner with an appropriate loss function to produce outcomes for different facial landmarks and full facial image and then ensemble them to derive the final result (hydrated/dehydrated). Section 6 presents the performance of our model. Our model outperforms the baseline solutions by a large margin. Since our model, trained and tested on a dataset for mild dehydrated conditions, can detect dehydration with reasonably good accuracy, it is expected that our model will lead to even more promising results for moderate or severe dehydration cases.

One of the major challenges we face is the lack of a publicly available dehydration dataset that fits our needs. To mitigate this issue, we build our own dataset consisting of 2340 sample pairs of images (hydrated-hydrated or hydrated-dehydrated) from 70 healthy volunteers. We do not include volunteers who have any other disease in the dataset to eliminate the inference of other diseases on the facial landmarks. Research [16,18] shows that people usually have mild dehydration during fasting. Therefore we consider the month of Ramadan for data collection when practising Muslims fast from sunrise to sunset. We develop a data collection app to capture the facial images of the volunteers in hydrated and dehydrated states. Then we apply our preprocessing techniques to remove the image noises, improve the image quality and extract the images of the facial landmarks. The landmarks that show the most prominent changes upon dehydration are chosen, including the lips, eyes, and surrounding regions. The preprocessed images of different landmarks constitute our final dataset on which

we train and test our model. Our data collection and preprocessing techniques are discussed in Sect. 3 and Sect. 4, respectively.

Finally, we develop a smartphone-based application, "Dehydration Scan" that captures the facial image of an individual using the integrated camera and classifies the condition of the individual as hydrated or dehydrated by running our developed model in the background. To the best of our knowledge, this is the first non-invasive approach for dehydration detection using a smartphone without requiring any additional equipment or expert skill while achieving adequate accuracy. In Sect. 7, we present our smartphone application to detect dehydration.

In summary, the contributions of this paper are as follows:

- We build a dataset that includes 2340 sample pairs of images from 70 individuals.
- We develop preprocessing steps to extract specific facial landmarks from the image frames and remove the noise associated with the effect of different lighting and background settings.
- We propose our siamese network-based contrast learning model to find the differences between a user's hydrated and dehydrated landmark images. We incorporate the individually derived values for different landmarks and the full facial image into a final score and classify the user's state as hydrated or dehydrated accordingly. We show the effectiveness of our model in experiments.
- We design and develop a complete and functional mobile application where users can take pictures of their faces and get to know their hydration status instantaneously.

2 Related Work

To date, existing works done on dehydration detection have been based on information collected and processed manually. In fact, no significant research has been done on automating this process. The existing methods require significant human intervention and often rely on a certain level of medical expertise. None of them focus on detecting dehydration using only mobile camera images. Table 1 shows a comparative analysis of existing dehydration detection techniques with ours in terms of different features. We observe that only our solution supports all desirable features.

The World Health Organization developed the Integrated Management of Childhood Illness (IMCI) algorithm [24] to guide health workers in detecting different levels of dehydration by monitoring clinical symptoms. Levine et al. [17] came up with the Dehydration: Assessing Kids Accurately (DHAKA) score for dehydration detection based on clinical symptoms. The authors conducted a study in Bangladesh, where local nurses analyzed children's dehydration status using both the DHAKA score and the IMCI algorithm. In a detailed comparison of the results, the DHAKA score was the more accurate predictor of dehydration for children. In [4], the authors developed a smartphone application that takes

Table 1. A Comparative Analysis of Existing Dehydration Detection Solutions

Paper	Technique	Health professional	Additional equipment	Additional cost	Time consuming
[4, 17, 24]	Clinical symptom monitoring	✓	✗	✓	✓
[18]	Skin conductance monitoring	✗	✓	✓	✗
[23]	Remote optical monitoring	✓	✓	✗	✗
[20]	Sweat electrolyte conductance monitoring	✗	✓	✓	✗
[25]	Oximeter signal monitoring	✗	✓	✓	✗
[19]	Image processing to detect skin mechanical properties through skin turgor test	✓	✗	✓	✓
Ours	Image processing using facial landmarks	✗	✗	✗	✗

clinical symptoms as input and then applies the IMCI algorithm of WHO to produce the result. This work aimed to check whether shifting from paper-based work to the smartphone app can improve the reliability and usability of the dehydration detection system.

The diagnostic technique proposed in [18] uses a non-invasive wearable sensor for collecting skin conductance data and detects dehydration based on the skin conductance state. Another study [10] examined the correlation between dehydration severity and the moisture level of oral mucous membrane measured through a moisture-checking device. In [33], the authors hypothesized whether dehydration can be detected using kinetic analysis of hemoglobin concentration.

In [23], the authors used a wristwatch to extract several bio-medical parameters. They implemented two optical approaches. One of them was the rotation of linearly polarized light by certain materials exposed to magnetic fields. Another was the extraction and separation of remote vibration sources. In [20], the authors used a conductometric sensor to measure sweat electrolyte conductance to detect dehydration. Another study [25] used photoplethysmographic signals with small, wearable pulse oximeters and set features based on the variable frequency complex demodulation. Those features were then fed to a support vector machine model to detect dehydration.

In [19], the authors used smartphones to capture the videos of skin turgor tests done on hands with two different methods - skin mark method and skin texture method. They ran image processing algorithms to extract skin mechanical properties and hydration levels from the frames of those videos. They used smartphones to track the turgor test's skin stretching and relaxation processes, which medical professionals usually undertake to check for dehydration.

3 Data Collection

At present, there exists no image dataset of people suffering from dehydration. Since our target is to detect dehydration in the early stages and people who come to the hospital are severely dehydrated in most cases, collecting data from hospitalized patients would not serve our purpose. To work around this problem, we leveraged the month of Ramadan, when practising Muslims fast from sunrise to sunset. Research [16,18] shows that people usually have mild dehydration during fasting, which does not result in severe health issues.

We developed a mobile application to achieve two goals:

1. To streamline the data collection process, and
2. To enable end-users to evaluate their hydration status instantaneously.

3.1 Data Collection Mobile Application

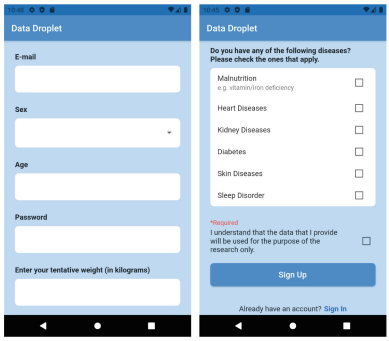
Our aim was to build an application that works on any mobile device irrespective of the operating system and configuration. Thus, we chose Flutter [30], a cross-platform application development framework, to create the smartphone app. We also used the Firestore Database and Storage services of Firebase [29] as the system's backend.

The app included five well-defined steps for every user to follow. Figure 1 shows a detailed view of the application interface in these steps.

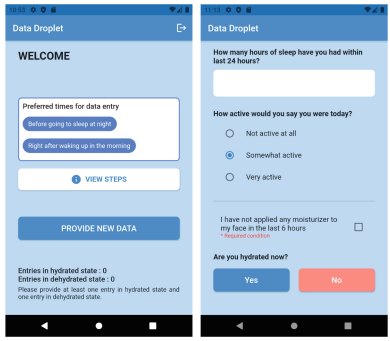
1. Users created a new account by providing relevant information, e.g., email address, age, sex, weight, and existing health conditions on the first usage. They were asked for explicit permission regarding the use of their data for our research before account creation. Afterward, they could log into their accounts from any smartphone device.
2. Every time users opted to provide data through the app, they had to give some basic information first, such as hours of sleep and activity level. Using moisturizer on the skin can neutralize the signs of dehydration almost entirely. So the app also required them to confirm that they had not applied any moisturizing product on their face in the last 6 h prior to providing the entry. We instructed the users to select hydrated state if they provide data at night after fluid intake and to select dehydrated state if they provide data during fasting.
3. Next, the users had to choose hydrated or dehydrated as their current state, which we later validated based on additional information. If hydrated, they were asked to mention their amount of fluid intake (in glasses) in the last six hours. Otherwise, they were asked to pick the approximate time of their last fluid intake. Based on the answer to the additional question, we verified the correctness of the user state (hydrated or dehydrated).
4. In the most crucial step, users recorded 5-second long videos of their faces using the front or back camera. The app displayed a bounding box around the users' faces in this process to ensure that no part of the face was outside

the camera view. They were also given the option to preview their captured face video and retake it if necessary before uploading it to Firebase cloud storage.

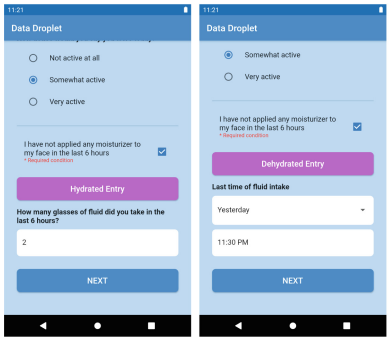
5. Upon successfully uploading a face video to Firebase cloud storage, the app stored the upload time against the user's credentials in its local storage. We implemented this additional check to disable the option for that user to give two data entries in less than 6 h gap.



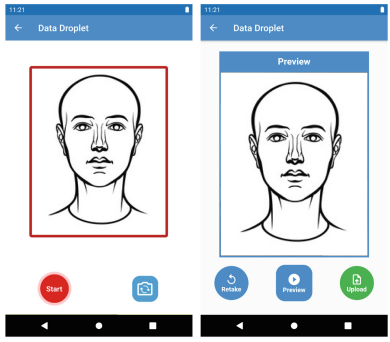
(a) Account creation with basic information and consent



(b) Basic information and regulation during data entry



(c) Additional information based on hydrated/dehydrated state (for dehydrated entry, the user can choose between today and yesterday and select the time from a time picker dialog)



(d) 5-second face video capture with preview and retake options (a dummy image of a human face has been shown to protect the privacy of the users who provided data)

Fig. 1. Various steps of the data collection portal of our smartphone application

The app supports a minimum SDK version of 21, which is the earliest release of the Android SDK that it can run on. It is equivalent to Android 5.0 (API level 21) or higher. The fundamental features and functionalities of Android

are available in this version and all subsequent versions. Since the latest stable version is Android 10.0 (API level 29), our application is runnable on a wide range of configurations. We used a few basic modules in our application - camera (for recording face videos), video player (for displaying a preview of captured video), and shared preferences (for locally storing entry timestamps) which work smoothly in most smartphones.

Table 2. Age distribution of dataset subjects

Age Range	No. of Participants
10–20	8
21–30	45
31–40	13
41–50	4

3.2 Dataset

We developed and distributed the mobile application described previously to build an in-house face video dataset explicitly suited to our use case. We prepared a final dataset containing 2340 pairs of images by strategically extracting and combining frames from the collected videos. We had a total of 70 volunteers in this process. 36 of them were male, and 34 were female. The participants were between 10 and 50 years old. However, many of them were undergraduate students, which added an age bias to our dataset. The age distribution is given in Table 2.

Initially, we had 326 face video entries from 70 individuals, which means that, on average, every user contributed 4–5 entries to our dataset. We collected videos instead of images because we could extract up to 10 frames from each video entry. To put it simply, we extracted a maximum of 10 hydrated or 10 dehydrated images of the same user from each video of that user. We discarded blurry or unusable frames using some preprocessing steps. Nevertheless, this approach helped us immensely to generate more data points for training and testing our model. Additionally, we checked if we had at least one hydrated and at least one dehydrated video entry from every user. Since we needed images of both conditions to create pair inputs for our model, we discarded the entries with no corresponding entry for the other condition. To ensure that our dataset does not reflect the symptoms of any disease or condition other than dehydration, we utilized the information gathered from the app. During account creation, the app displayed a list of diseases (e.g., malnutrition, heart diseases, kidney diseases, diabetes, skin diseases, sleep disorder) that can interfere with the signs of dehydration. From the list, users selected any condition that applied to them. While preparing our image dataset, we excluded all users suffering from any of the mentioned diseases. This filtering was necessary to prevent our model from falsely identifying symptoms of those diseases as effects of dehydration.

Ground Truth Validation. In the case of dehydrated data entries, we collected the number of hours passed after the last fluid intake as additional information from the user and used it to validate the given hydration status label. We found 11 video entries for which this time gap was less than 6 h. Thus, we discarded those noisy entries considering that the users might not be in a dehydrated state in those and proceeded with the remaining 315 entries. The average number of hours after the last fluid intake in the selected entries was 12 h. On the other hand, for hydrated entries, we collected the amount (in glasses) of fluid consumed in the last 6 h as additional information to make sure that the user is in a hydrated state. The average glass count was 3.

For every user, we selected one of the hydrated images as the reference image for that user. Then we created all possible combinations by pairing the reference image with every available image (hydrated/dehydrated) of that user. While generating these combinations, we did not consider the creation time or sequence of a particular entry. The only condition in pairing any hydrated or dehydrated image with the reference image was that the two images belonged to the same user. We conducted this step of the process based on the assumption that for any particular user, the effects of dehydration on different facial regions would be pretty similar regardless of the time he/she captured the condition. In this step, we significantly augmented the face video dataset collected through our app and prepared a final dataset of 2340 pairs of images. The dataset is balanced, consisting of 1170 reference-hydrated and 1170 reference-dehydrated pairs of samples.

4 Data Preprocessing

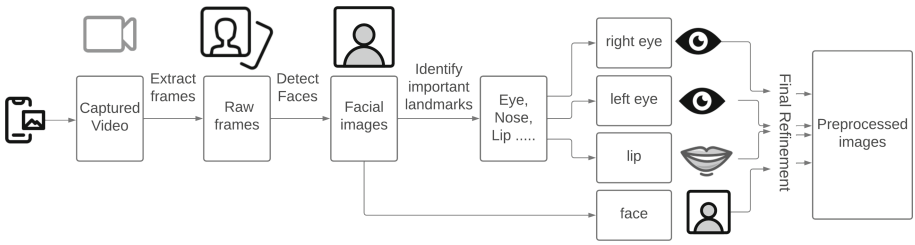


Fig. 2. Preprocessing steps

In this section, we elaborate on the automated preprocessing steps performed before we feed our data into the model for training and prediction. Our model is at its core a siamese network that uses contrastive loss in order to learn and detect the differences between a hydrated and a dehydrated image. Thus, we need to provide pairs of images of the same individual to the model during training. The model calculates a similarity score for the new image to predict its hydration level using the trained weights.

From the data collection portal of our app, we obtain 10-second long videos of both hydrated and dehydrated states of each user. These videos have a lot of noise, such as varying levels of light in the image, differences in resolution, contrast, saturation, and overall image quality due to variation in image sensors. We need to eliminate these adverse effects and format the final image to emphasize the maximum contrast between the hydrated and dehydrated states and offset the other differences. Since our model trains with images, the first step is to extract individual frames from the videos. We do this using the python library OpenCV. Using the variance of the Laplacian, we filter out the blurry images and keep the ones that report the highest values of sharpness. We select only the images that cross a certain variance threshold and discard the rest. A final manual pass is done to ensure that no anomalous images have spilled through. An overview of the preprocessing steps is presented in Fig. 2.

Our final model uses the similarity scores of facial landmarks as different features and creates an ensemble by assigning different weights to each of these scores. We determine the combination of weights for which our model provides the best performance in experiments. The values of the weights that our model uses for different landmarks are specified in Sect. 6.1. We train individual models for each of the landmarks separately. Thus, it is paramount for us to identify, detect and segregate the landmarks in our preprocessing step. Since signs of dehydration are primarily visible in areas surrounding the eyes, the skin, and lips, we particularly emphasize our landmarks in those areas. For the eyes, we include areas around the eyelids as well as the under-eye region since those are the areas most affected. The entire face image is passed on as a separate landmark as well.

We use the Haar Cascade classifier included in OpenCV to extract the landmarks from images. The classifier returns four coordinates for each of our individual landmarks. Each of the four coordinates represents a corner of a bounding box surrounding the landmark. Joining the four coordinates, we obtain a bounding box locating the position of the landmark. We then separate the landmarks by cropping them out from the original image. Lastly, we scale the images to our required pixel size of 200*200 before feeding them into the model. The pixel size is chosen to strike the optimal balance between retaining the highest possible level of details while keeping it lightweight enough for on-device inference.

5 Model Overview

Unlike most facial image classification problems, which are typically solved by utilizing model-extracted features, dehydration detection is incredibly challenging because the symptoms of mild dehydration are tough to decipher. Therefore, simply attempting to capture facial features with a traditional deep learning model would not perform well in this particular problem.

We extract and prepare images of specific facial landmarks in the preprocessing step, which is quite crucial to make it easier for our model to capture differences down to the tiniest detail. Our system requires only one reference image

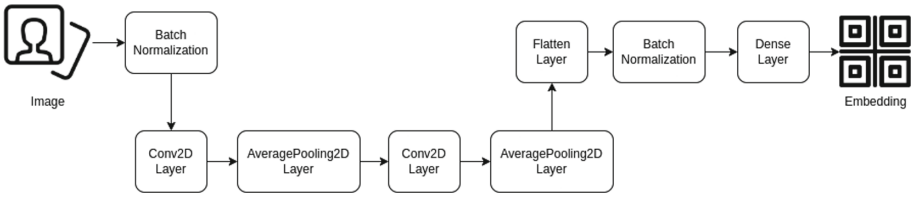


Fig. 3. Layered view of base convolutional network

taken in the hydrated state from every individual. Thus, we build a model that can predict a user's hydration status from a pair of images - one of them is the initial reference image captured in the hydrated state, and the other is an image taken at any hydration level at any moment. To serve this purpose, we leverage an artificial neural network model called siamese network [5], specifically designed for use cases similar to ours. Koch et al. [15] used siamese neural networks to address the one-shot classification problem where predictions need to be made from a single instance of any class. The way we structure our approach is quite similar to this problem. Given a reference hydrated image of any user, our model processes any input image of the same user, evaluates its similarity to the reference, and classifies it as hydrated or dehydrated. Since the model only requires a single pair of images from any user, its prediction technique is comparable to one-shot learning.

The siamese network-based architecture primarily comprises two identical towers or sister networks with shared weights. Each takes one image as input and generates a feature embedding for that image.

In Fig. 3, we can see a detailed graphical overview of our convolutional network. Each network is essentially a two-dimensional convolutional neural network containing the same sequence of layers. The input images first go through batch normalization before getting processed by the convolutional networks. Our model constructs the networks from consecutive 2D convolutional and average pooling layers, followed by batch normalization and a dense layer with rectified linear unit (ReLU) activation. Our model then converts the generated feature embeddings into a single merged layer by computing their cosine distance.

$$similarity(A, B) = \frac{A \cdot B}{\|A\|_2 \times \|B\|_2} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

As can be seen from Eq. 1, the cosine similarity of two vectors A and B is simply the normalized dot product of A and B. Here, $\|A\|_2$ is the L2 norm of vector A. A high cosine similarity implies that the images closely match each other and have minimal contrast between them.

$$distance(A, B) = 1 - similarity(A, B) = 1 - \frac{A \cdot B}{\|A\|_2 \times \|B\|_2} \tag{2}$$

From Eq. 2, we can see that cosine distance and cosine similarity are complementary measures. Two identical vectors with an angle of zero degrees between them have a similarity score of 1 and a distance of 0.

Using other distance metrics such as Euclidean distance and Manhattan distance in the merging step does not give promising results. The Euclidean distance between two vectors refers to their straight-line distance, while the Manhattan distance refers to the sum of the distances along each axis. Unlike these measures, cosine distance depends on the angle between two vectors and does not consider the size of the vectors. Thus, in our case, cosine distance proves to be a better metric for differentiating the embeddings. The distance values that signify the contrast between the input images are passed to the final neural network consisting of batch normalization and a typical dense layer with sigmoid activation. To summarize, our proposed contrast learning model takes pairs of images belonging to the same user as input, estimates their similarity, and produces a score between 0 and 1 accordingly. The architecture of this model is illustrated in Fig. 4.

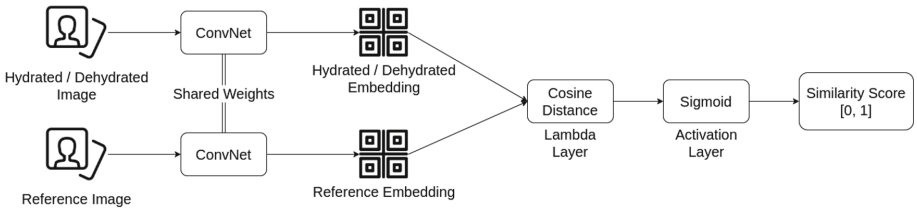


Fig. 4. Siamese-network based contrast learner

Another critical aspect of the model is the choice of the loss function, which is used in the backpropagation to update the weights of the convolutional layers. Two loss functions are generally used in siamese networks - triplet loss and contrastive loss [13]. If triplet loss is used, the model takes in three inputs - one reference image, one similar or neighbor image, and one dissimilar or distant image. The key idea in this approach is to minimize the reference-neighbor distance and maximize the reference-distant distance. Contrastive loss, on the other hand, deals with pairs of images. The pairs can be reference-neighbor or reference-distant, which is the exact format of the samples in our dataset. Thus, we choose contrastive loss as the loss function for our model.

$$L = Y \times D^2 + (1 - Y) \times \max(margin - D, 0)^2 \tag{3}$$

In Eq. 3, the contrastive loss L is computed from the predicted value Y, the cosine distance D, and the baseline distance $margin$ for which the model should classify pairs as dissimilar. The default value for the $margin$ is 1.

The contrast learner is run parallelly for the image pairs of each of our selected facial landmarks - the right and left eyes, lip, and nose. We acquire a similarity score from every execution of the model and calculate a weighted sum of those scores to derive the final prediction. The weights for the landmarks are obtained

through an extensive trial and error process. After adequate experimental analysis, we find nearly optimal weights to account for the relevance of each landmark in maximizing the prediction accuracy. We also run the model on entire face images extracted from video frames. In short, to reach the conclusive prediction of our system, we take into account every essential region of the face, do individual similarity estimations using our siamese network-based model, and combine them to classify the input image as hydrated or dehydrated. Figure 5 shows an overview of our proposed model.

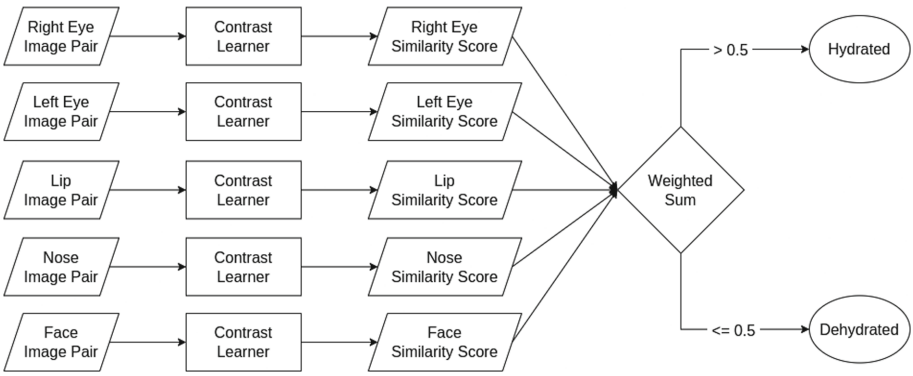


Fig. 5. Classification using a weighted sum of similarity scores from image pairs of individual landmarks

There are two core factors behind the considerable prediction accuracy of our proposed model.

- We take a personalized approach instead of generalizing the problem. We focus on the contrast between hydrated and dehydrated images of the same person, which helps us cut out the differentiating factors among separate individuals.
- We run our model separately on the segregated and preprocessed images of various facial regions to better capture the changes occurring in each region. This approach is particularly substantial for identifying mild dehydration since the symptoms might not appear in the same region for every individual.

One additional concern in the development of our model is handling the resource constraints of smartphones. In order to predict hydration status using the limited processing power of smartphones, we choose TensorFlow as our artificial intelligence framework and devise a sufficiently lightweight model. We convert our TensorFlow model to a TensorFlow Lite model, which is smaller, significantly faster, and runs more efficiently on a mobile processor. Our smartphone application runs the lite model in the background whenever a user submits an input image of his/her face, performs on-device computation, and delivers instantaneous results - hydrated or dehydrated.

6 Performance Analysis

In this section, we present the performance of our proposed dehydration detection model.

6.1 Experimental Setup

Dataset and Preprocessing: We train and evaluate our model on our procured dataset elaborated in Sect. 3. The data first passes through some preprocessing steps as discussed in Sect. 4 before being split into train, test, and validation sets. As mentioned in Sect. 3.2, our dataset contains 2340 pairs of images, where 1170 pairs contain two hydrated images, and the other 1170 pairs include one hydrated and one dehydrated image. We take 60%, 20%, and 20% of the pairs as train, validation, and test sets, respectively. We use samples of three separate sets of users for training, validating, and testing datasets. The train, validation, and test split is discussed more elaborately in Sect. 6.6.

We calculate all performance metrics presented in this section on our test set, which is completely isolated from the training phase with no peeking involved anywhere in the pipeline.

Parameter Settings: We select a batch size of 16 and run the model for 10 epochs. The number of epochs is chosen through experimentation as elaborated in Sect. 6.5. The image size is selected to be 200*200 pixels in order to capture as much detail as possible while keeping it reasonably less resource-heavy. For the final ensemble, we choose weights for each of the individual landmarks as follows: Left and Right Eye: 0.275, Lip: 0.225, Nose: 0.125, Entire Face: 0.1.

Evaluation Criteria: We choose Accuracy, Specificity, Recall, and F1 score as our evaluation criteria. Accuracy measures how many hydrated and dehydrated images are correctly classified as hydrated and dehydrated, respectively. Specificity measures out of all the dehydrated images, how many are correctly classified as dehydrated. Recall tells us how many are correctly classified as hydrated from all the hydrated images. To understand the F1 score, we first need to know what precision is. Precision measures how many images that are classified as hydrated are actually hydrated. F1 score is the harmonic mean of precision and recall.

6.2 Comparative Analysis with Baselines

We consider the following four baselines and compare the performance of our solution with them.

- **Basic CNN:** For our basic CNN structure, we have used one convolutional layer followed by a pooling layer and a dense layer at the end. This is representative of the most rudimentary implementation of an image classifier.
- **VGG16** [26]**:** VGG was invented with the purpose of enhancing classification accuracy by increasing the depth of the CNNs. VGG 16 has 16 weight layers and is used for object recognition.

- **Xception** [6]**:** Xception is a convolutional neural network architecture consisting of 71 layers. This network gives a rich representation of images when trained on an expansive dataset.
- **Resnet50** [14]**:** Resnets are neural networks that use skip connections to solve the problem of vanishing gradients. Resnet50 is a Resnet that is 50 layers deep.

For VGG16, Xception, and Resnet50, we use transfer learning by initializing the models with weights trained on imagenet [8]. After that, all the baselines have been run on our dataset end-to-end to come up with the final predictions. The performance metrics are then compared with those of our proposed solution. We can get an overview of the performance comparison between the baselines and our model from Table 3.

Table 3. Performance comparison of our proposed model with different baselines

Model	Accuracy	Specificity	Recall	F1 score
Basic CNN	51.1%	15.2%	48.4%	56.6%
VGG16	34.4%	31.9%	52.3%	52.6%
Xception	42.5%	17.6%	38.7%	47.1%
ResNet50	65.9%	23.5%	99.7%	79.3%
Our Model	76.1%	52.1%	99.1%	80.7%

It is apparent from the above data that our proposed siamese network-based model provides a better classification accuracy, specificity, recall, and F1 score than those of the baselines. The key differentiating factor here is the emphasis of our model on detecting contrasts between the hydrated and dehydrated states. To assist the contrast detection, we concentrate on areas of the face that show the most significant changes during dehydration. Our model can learn differences in facial features upon dehydration better than the conventional models, which look at the face image as a whole and do not prioritize specific landmarks like our model.

6.3 Effect of Landmarks

To make the contrast detection between hydrated and dehydrated states more robust, we identify the landmarks that show changes upon dehydration. We now discuss how each of these individual landmarks performs on its own to detect dehydration. The performance comparison of how each of the landmarks performs separately is presented in Table 4.

The left and right eyes show the best performance out of all the landmarks. This is expected since one of the main symptoms of dehydration is sunken eyes. This is consistent with the studies made in [28] where it is mentioned that

Table 4. Performance comparison between individual landmarks

Landmark used	Accuracy	Specificity	Recall	F1 score
Entire Face	58.8%	20.4%	97.2%	70.2%
Nose	59.7%	19.5%	99.7%	71.3%
Lip	61.9%	29.3%	94.4%	71.2%
Right eye	76.1%	52.1%	96.3%	80.7%
Left eye	72.3%	69.3%	75.3%	73.1%
Left eye flipped	74.3%	50.6%	97.9%	79.2%

dehydration symptoms like sunken eyes, undereye darkness, or discoloration can show up faster because the skin in this region is thinner than other body parts. To properly capture this, we took into account the regions surrounding the eyes during our initial crop. The next best landmark is the lip, most likely due to the flakiness or dryness visible when there is no fluid intake for an extended period. [9] cites dry lips as one of the leading signs of dehydration. The entire face image gives the lowest accuracy, which further proves why conventional models do not perform well in this scope. The entire face data taken together do not provide enough information for classifying a dehydrated image from a hydrated one.

Additionally, we flipped left eye images horizontally to make them similar to right eye images. We included this step to make all the eye images more homogeneous and minimize the unexpected performance gap between left and right eye images.

6.4 Ablation Study

We perform an ablation study by removing one landmark at a time and performing the classification. We intend to see the effect of each of the landmarks on the final classification. The results of the ablation study are summarized in Table 5.

It is evident from the data that excluding any landmark reduces the performance of our model; for example, the obtained accuracy for excluding any

Table 5. Ablation Study

Landmark excluded	Accuracy	Specificity	Recall	F1 score
Entire Face	70.6%	45.7%	95.4%	76.4%
Nose	70.9%	41.9%	95.6%	77.5%
Lip	68.3%	36.7%	94.3%	75.9%
Right eye	64.4%	28.8%	92.4%	73.8%
Left eye	65.7%	44.7%	97.8%	78.3%
Left eye flipped	65.3%	30.6%	95.8%	74.2%

landmark is less than the overall accuracy 76.1%. Thus, each of the landmarks that we use as model features contributes to improving the prediction. The effect of removing the eyes from the classification has the highest impact. This is followed by the removal of the lips. This is consistent with our assumption that the eyes and lips show the most visible changes during dehydration.

6.5 Effect of Number of Epochs

In this experiment, we choose the optimal number of epochs to get the best performance of the model. Usually, up to a certain threshold, the more epochs the model runs for, the better the accuracy and performance measures are. After a point, the model starts overfitting to the training set, and its performance starts degrading. We need to find the optimal epoch number that provides the best results without introducing overfitting. Table 6 and Fig. 6 give us an overview of the performance metrics by varying the number of epochs.

Table 6. Performance comparison between varying Epoch numbers

No. of Epochs	Accuracy	Specificity	Recall	F1 score
5	68.1%	36.3%	95.7%	75.8%
8	71.6%	43.3%	97.8%	77.9%
10	76.1%	52.1%	99.1%	80.7%
12	66.7%	39.1%	94.4%	73.9%

The data shows that the model reaches maximum performance at 10 epochs. After that, the performance slowly starts degrading as it starts introducing overfitting. We, therefore, stopped the model after 10 epochs during our final classification.

6.6 Effect of Train-Test Ratio

In this section, we present our experiments to choose the optimal split for the test, validation, and train sets. On the one hand, increasing the train set size gives our model more data to train on, essentially boosting the performance. However, keeping little data on the validation set runs the risk of overfitting the train set. To find the optimal balance, we conduct multiple runs with different ratios of the split as presented in Table 7.

As we can infer from the data, the optimal performance is achieved with a 60-20-20 split of train, validation, and test set.

6.7 Male vs Female Participants

In this section, we compare the performance metrics of male and female participants. The findings are presented in Table 8. The results for male participants

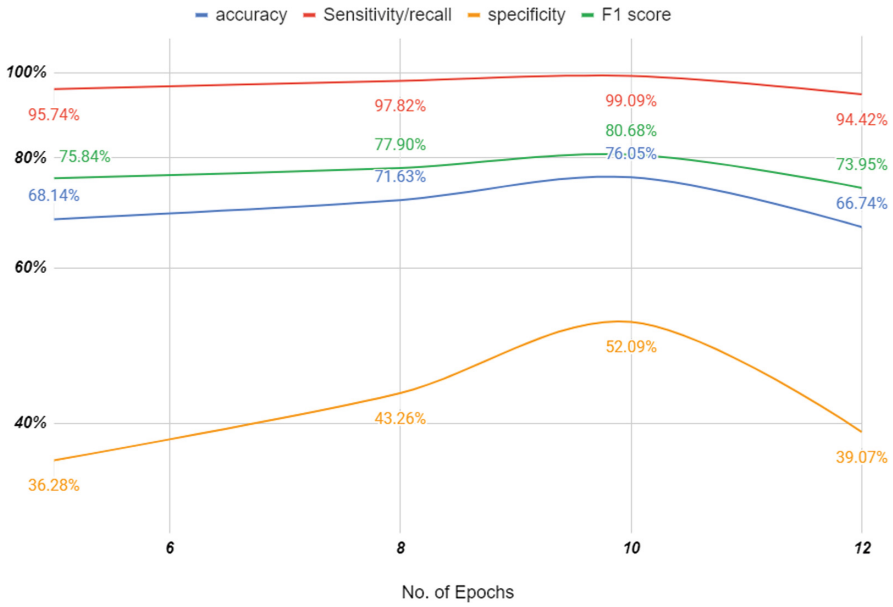


Fig. 6. Log graph of performance comparison between varying epochs

Table 7. Performance comparison between varying Train-Test ratios

% Train	% Validation	% Test	Accuracy	Specificity	Recall	F1 score
70%	15%	15%	62.1%	30.9%	93.4%	71.1%
70%	10%	20%	69.7%	48.1%	91.3%	75.1%
80%	10%	10%	54.3%	37.7%	71.1%	60.8%
60%	20%	20%	76.1%	52.1%	99.1%	80.7%

were derived by training and testing our model only on male data. We repeated the same process for female participants.

Although the accuracy is quite similar in both cases, there is a noticeable difference in the specificity and recall values. This indicates that our model performance might vary across genders.

6.8 Effect of Multiple vs Single Reference

Here we compare the performance of our model using single and multiple reference images. In the case of a single reference image, we paired the reference with each hydrated and each dehydrated image of the same individual. Similarly, in the case of multiple reference images, we selected multiple hydrated images for a user based on their sharpness and paired each of the references with each remaining hydrated and dehydrated image. The performance comparison is shown in Table 9.

Table 8. Performance comparison between male and female participants

Participant Gender	Accuracy	Specificity	Recall	F1 score
Male	72.9%	64.9%	99.8%	78.7%
Female	72.1%	83.6%	60.7%	68.5%

Table 9. Performance comparison between single and multiple reference image

No. of references	Accuracy	Specificity	Recall	F1 score
1	76.1%	52.1%	99.1%	80.7%
2	72.2%	44.4%	99.7%	78.3%
3	68.4%	36.8%	99.8%	75.9%

It is evident from the experiment results that using multiple images as references does not improve our model performance. One possible reason behind this might be that the hydrated frames of an individual are very similar to one another. Therefore, multiple reference images captured in hydrated conditions only increase the dataset size without contributing any new, substantial information regarding dehydration. Besides, the model might overfit the data of the users in the train set and, consequently, give a poor performance for face images of unseen test subjects.

6.9 Cross Validation

We partitioned the data into 5 non-overlapping folds or subsets and performed cross-validation by running our model 5 times. Every time, we chose a different subset for testing and used the remaining data for training and validation. The results from the different runs are shown in Table 10 along with the average performance.

Table 10. 5-fold cross-validation results

Test set	Accuracy	Specificity	Recall	F1 score
1	69.4%	59.7%	78.9%	72.1%
2	69.7%	45.6%	93.9%	75.6%
3	76.1%	64.9%	87.2%	78.5%
4	73.7%	47.4%	99.7%	79.2%
5	76.1%	52.1%	99.1%	80.7%
Average	73.0%	53.9%	91.8%	77.2%

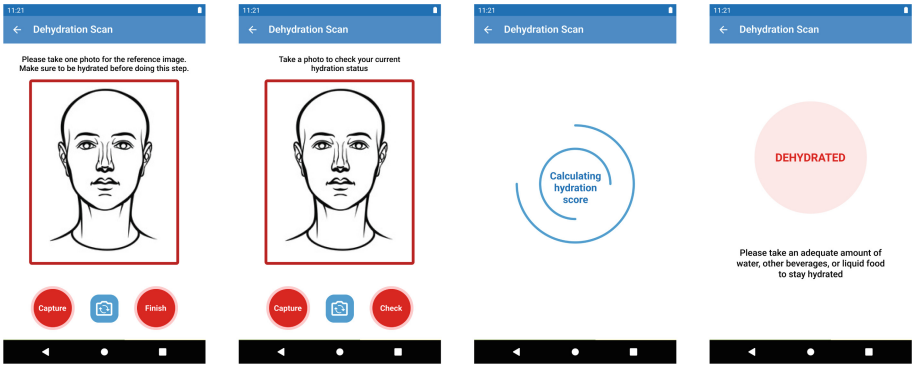
7 "Dehydration Scan" App Overview

In this section, we present the interface and functionality of our smartphone application, "Dehydration Scan". The app has been designed to minimize the

number of steps that a user must complete to check his/her hydration status. The user interface is also relatively simple and highlights the required steps. We share a detailed view of the interface in Fig. 7.

While using the app, a typical user performs the following actions.

1. First, a user needs to take a picture in hydrated condition to start using the app. This picture is saved in the local storage of the app as the reference image for future predictions. Since the facial condition of the user may change over time because of factors like age and weather, the app provides an option to retake and update the reference image at any point in time. Besides, it is crucial for the user to be adequately hydrated while capturing the reference image. Otherwise, our dehydration detection model will fail to deliver accurate predictions.
2. Afterward, the user only needs to provide a face image to check for dehydration. This image goes through the preprocessing steps mentioned in Sect. 4 and gets paired with the previously captured and stored reference image to prepare the input for the model.
3. Then the app runs the trained lite version of our proposed model on the pair of images. This computation takes place entirely on-device and does not require any cloud storage or services. Since the model has already been trained on our dataset, it does not require any additional training time and executes in a matter of seconds.
4. After calculating the final score, the app displays the predicted class of the condition of the user.



(a) Taking a reference image (updatable) from the user for future evaluations

(b) Taking image input to pair it up with the reference image for comparison

(c) Running the on-device model to compute hydration score for the input image

(d) Displaying the final prediction to the user - either Hydrated or Dehydrated

Fig. 7. Steps of dehydration detection using Dehydration Scan

We use the TensorFlow Lite library to run our model in the background, and the minimum SDK version for this library is 21. Therefore, like our data collection application described in Sect. 3.1, Dehydration Scan also operates well in android devices with API level 21 or higher. This range of configurations covers almost all available android smartphones. Moreover, Dehydration Scan is a standalone smartphone application that does not require an internet connection to function. Thus, it serves to be a ubiquitous solution to the dehydration detection problem.

8 Limitations and Future Direction

While we demonstrate the strong potential of our proposed siamese model, there are scopes to improve it even further. Following is a list of limitations we identified from our research and the corresponding future research directions one might pursue:

- **Mitigating the absence of a publicly available diversified dataset:** There is an acute deficiency of a publicly available diversified dataset in the domain of dehydration detection. Although we have managed to procure a decent dataset of our own, it lacks diversity in ethnicity, age, and other aspects. Also, the dataset we worked on is imbalanced in terms of the age range of the participants. There are very few data points over the age of 40. Our dataset is primarily composed of undergraduate students and, as such, introduces an age bias. Overall there is massive scope for a better, more robust dataset. Improving the dataset can also improve the classification performance.
- **Automating the assignment of weights to the different landmarks:** In our implementation, we have manually assigned weights to the different landmarks during the computation of the final prediction. We have chosen the weights that provide the overall best performance over the entire test set through trial and error. However, the optimal weights may vary for each individual, so a generalized assignment of weight for everyone may not give the best results.
- **Using more sophisticated models with higher parameter counts:** As smartphone hardware gets more powerful year after year, it becomes feasible for more complex and sophisticated algorithms to be run on-device on mobile processors. The limitations in hardware capabilities will not be present in a few years when more powerful smartphones penetrate the general mass. Therefore, future research should concentrate on more sophisticated models that give higher prediction accuracy, albeit with more resources.

9 Conclusion

We developed a non-invasive smartphone-based dehydration diagnostic solution that does not need additional cost, hardware, or skilled workers and produces

results instantaneously. Since our approach delivers acceptable accuracy (on average 76.1%) for mild to moderate dehydration, we can infer that more severe cases of dehydration can be predicted with a higher degree of confidence. Our developed siamese network-based dehydration detection model outperforms the baseline models by a large margin (i.e., our model achieves at least 10% more accuracy and 20% more specificity than those of the baseline models). Experimental results also show that eyes are the most affected facial landmark due to dehydration. Our smartphone-based dehydration diagnosis tool may not achieve the accuracy level of the clinical diagnosis by professional medical equipment or expert personnel. However, in most cases, our solution can provide crucial insights and prompt the users to take necessary actions early on. We aim to extend our solution to detect different dehydration levels (e.g., mild, moderate, severe) in the future.

Acknowledgements. This research has been done at Bangladesh University of Engineering and Technology (BUET). We thank all the volunteers who participated with their valuable data to make this research possible.

References

1. Ahmed, S.M., Hossain, M.A., RajaChowdhury, A.M., Bhuiya, A.U.: The health workforce crisis in Bangladesh: shortage, inappropriate skill-mix and inequitable distribution. Hum. Resour. Health **9**(1) (2011). https://doi.org/10.1186/1478-4491-9-3
2. Armstrong, L.E., et al.: Mild dehydration affects mood in healthy young women. J. Nutr. **142**(2), 382–388 (2012)
3. Barrell, A.: What to know about dehydrated skin (2021). https://www.medicalnewstoday.com/articles/dehydrated-skin#causes. Accessed 14 Feb 2022
4. Bilal, S., et al.: Evaluation of standard and mobile health-supported clinical diagnostic tools for assessing dehydration in patients with diarrhea in rural Bangladesh. Am. J. Trop. Med. Hyg. **99**(1), 171–179 (2018)
5. Chicco, D.: Siamese neural networks: an overview. In: Cartwright, H. (ed.) Artificial Neural Networks. MMB, vol. 2190, pp. 73–94. Springer, New York (2021). https://doi.org/10.1007/978-1-0716-0826-5_3
6. Chollet, F.: Xception: deep learning with depthwise separable convolutions, pp. 1800–1807 (2017). https://doi.org/10.1109/CVPR.2017.195
7. Dehydrated? How not drinking enough water impacts your eyes (n.d.). https://www.essilorusa.com/newsroom/dehydrated-how-notdrinking-enough-water-impacts-your-eyes. Accessed 14 Feb 2022
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
9. Dry lips (n.d.). https://www.healthline.com/health/dehydration-white-tongue#other-symptoms. Accessed 29 Apr 2022
10. Fukushima, Y., et al.: A pilot clinical evaluation of oral mucosal dryness in dehydrated patients using a moisture-checking device. Clin. Exp. Dent. Res. **5**(2), 116–120 (2019)

11. Gairola, S., et al.: SmartKC: smartphone-based corneal topographer for keratoconus detection. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(4) (2022)
12. Ganio, M.S., et al.: Mild dehydration impairs cognitive performance and mood of men. Br. J. Nutr. **106**(10), 1535–1543 (2011). https://doi.org/10.1017/s0007114511002005
13. Ghojogh, B., Sikaroudi, M., Shafiei, S., Tizhoosh, H.R., Karray, F., Crowley, M.: Fisher discriminant triplet and contrastive losses for training Siamese networks. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE (2020). https://doi.org/10.1109/ijcnn48605.2020.9206833
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
15. Koch, G.R.: Siamese neural networks for one-shot image recognition (2015)
16. Leiper, J.B., Molla, A.M.: Effects on health of fluid restriction during fasting in Ramadan. Eur. J. Clin. Nutr. **57**(S2), S30–S38 (2003). https://doi.org/10.1038/sj.ejcn.1601899
17. Levine, A.C., et al.: External validation of the DHAKA score and comparison with the current IMCI algorithm for the assessment of dehydration in children with diarrhoea: a prospective cohort study. Lancet Glob. Health **4**(10), e744–e751 (2016). https://doi.org/10.1016/s2214-109x(16)30150-4
18. Liaqat, S., Dashtipour, K., Arshad, K., Ramzan, N.: Non invasive skin hydration level detection using machine learning. Electronics (Basel) **9**(7), 1086 (2020)
19. Liu, C., Tsow, F., Shao, D., Yang, Y., Iriya, R., Tao, N.: Skin mechanical properties and hydration measured with mobile phone camera. IEEE Sens. J. **16**(4), 924–930 (2016)
20. Liu, G., Smith, K., Kaya, T.: Implementation of a microfluidic conductivity sensor—a potential sweat electrolyte sensing system for dehydration detection. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, pp. 1678–1681. IEEE (2014). https://doi.org/10.1109/EMBC.2014.6943929
21. Mariakakis, A., Banks, M.A., Phillipi, L., Yu, L., Taylor, J., Patel, S.N.: Biliscreen: smartphone-based scleral jaundice monitoring for liver and pancreatic disorders. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(2) (2017). https://doi.org/10.1145/3090085
22. Mariakakis, A., et al.: Pupilscreen: using smartphones to assess traumatic brain injury. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(3), 81 (2017). https://doi.org/10.1145/3131896
23. Ozana, N., et al.: Improved noncontact optical sensor for detection of glucose concentration and indication of dehydration level. Biomed. Opt. Express **5**(6), 1926–1940 (2014)
24. Perkins, B.A., et al.: Evaluation of an algorithm for integrated management of childhood illness in an area of Kenya with high malaria transmission. Bull. World Health Organ. **75**(Suppl. 1), 33–42 (1997)
25. Reljin, N., et al.: Automatic detection of dehydration using support vector machines. In: 2018 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade. IEEE (2018)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
27. Skin turgor (n.d.). https://medlineplus.gov/ency/article/003281.htm. Accessed 14 Feb 2022

28. Sunken eyes (n.d.). https://www.healthline.com/health/dry-eye/ask-the-expert-dry-eye-dehydration#hydrating. Accessed 29 Apr 2022
29. Wikipedia contributors. Firebase—Wikipedia, the free encyclopedia (2021). https://en.wikipedia.org/w/index.php?title=Firebase&oldid=1054631697. Accessed 15 Feb 2022
30. Wikipedia contributors. Flutter (software)—Wikipedia, the free encyclopedia (2022). https://en.wikipedia.org/w/index.php?title=Flutter_(software)&oldid=1070411100. Accessed 15 Feb 2022
31. Wikipedia contributors. Siamese neural network—Wikipedia, the free encyclopedia (2021). https://en.wikipedia.org/w/index.php?title=Siamese_neural_network&oldid=1020522415. Accessed 15 Feb 2022
32. Xu, X., et al.: Listen2cough: leveraging end-to-end deep learning cough detection model to enhance lung health assessment using passively sensed audio. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(1), 42:1–42:22 (2021). https://doi.org/10.1145/3448124
33. Zdolsek, J., Li, Y., Hahn, R.G.: Detection of dehydration by using volume kinetics. Anesth. Analg. **115**(4), 814–822 (2012)