Character Recognition Architecture for Low-cost Glucometers: A step towards the Deployment of an eHealth System in Colombian Rural Communities

Cristian Camilo Gil Sánchez
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Universidad Antonio Nariño
Bogotá, Colombia
ORCID: 0009-0006-2362-7684
cgil86@uan.edu.co

Luis Alberto Cruz Salazar
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Universidad Antonio Nariño
Bogotá, Colombia
ORCID: 0000-0001-8386-5568
luicruz@uan.edu.co

David Santiago Delgadillo Leguizamón
Facultad de Ingeniería Industrial
Universidad Antonio Nariño
Bogotá, Colombia
ORCID: 0000-0002-2328-5200
ddelgadillo142@uan.edu.co

Leonel Paredes-Madrid
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Universidad Antonio Nariño
Bogotá, Colombia
ORCID: 0000-0001-8609-0681
paredes.leonel@uan.edu.co

Fabian Mauricio León Vargas
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Universidad Antonio Nariño
Bogotá, Colombia
ORCID: 0000-0002-1839-2036
fabianleon@uan.edu.co

***Abstract*—This work presents a digit recognition algorithm designed for low-cost glucometer displays, leveraging the pattern retrieval capabilities of Hopfield and Convolutional Neural Networks. The system processes input images through grayscale conversion, adaptive binarization, and morphological filtering to enhance digit visibility. Each digit is segmented, resized, and transformed into a binary vector suitable for associative memory recognition. A pre-trained Hopfield network retrieves the closest stored pattern by converging to a stable state, effectively identifying each individual digit; this output is then sourced to a Convolution Neural Network that let us assemble meaningful data matching for glucometric information, date and hour of measurement. The system was modeled by Unified Modeling Language diagrams with a pseudocode and successfully tested on Glucoquick glucometers, which are popular in rural areas of Colombia, under conditions of daylight and artificial lighting.**

Keywords—Diabetes treatment, Convolutional Neural Network, Hopfield Neural Network, Image segmentation, Glucometer, Unified Modeling Language, UML.

I. Introduction

Diabetes is a chronic disease with a relatively high prevalence in rural communities of Colombia [1]. High-carb diets are usually responsible for the development of Type 2 Diabetes (T2D) in adults of 50 years old and over [2]; Inhabitants of rural communities in Colombia are not the exception to this rule. As a part of blood glucose control in T2D patients; regular measurements of glycemia are required in order to ensure the proper amount of oral intake medication and/or insulin shots; this process is the typical self-care routine of diabetic patients that is regularly supervised by a Specialist Doctor (SD).

For this purpose, the SD requires a detailed record of patient’s blood glucose over the last month, or since the last medical visit. Based upon medical instructions, up to ten daily reading of blood glucose may be required for patients with comorbidities or to those with Type 1 Diabetes (T1D), but usually, a couple readings a day are more than enough for the SD to trim the daily dosage of oral medication and/or the insulin shots per day.

Fortunately, in recent years, software solutions have been developed for diabetic patients that assist them in the aforementioned daily self-care routine [3]; this is the case of platforms such as Tidepool [4], Glooko [5] and Nightscout [6]. These software solutions allow diabetic patients to upload their glycemia readings to a virtual user profile which is regularly checked by the SD, thus enabling a timely update in oral medication and/or insulin shots via direct messaging between the doctor and the patient.

The main innovation introduced by these solutions is the ability to automatically extract the information recorded by the glucometers using a USB or Bluetooth interface; this is a desirable, time-saving feature with multiple advantages, such as: (1) no paper registration of blood glucose is required by the patient, (2) exact date and time are recorded by the glucometer, (3) connection with other type of health monitors and fitness gadgets is also possible, e.g. blood pressure, Apple Watch or similar, and (4), continuous glucose monitoring (CGM) is also supported; this is especially important for T1D patients.

Unfortunately, only a few brands of commercially available glucometers in Colombia support either Tidepool, Glooko, or Nightscout platforms; this the case of the Accu-Chek Aviva [7] and the One Touch Ultra [8] glucometers. Popular brands of glucometers such as Medtronic, Abbot, Glucocard and Insulet are supported by the aforementioned

This research was funded by MinCiencias (Colombia) through grant number 1233949110684

platforms, but are not commonly found in the country; this condition sets a clear technological gap for the deployment of eHealth systems aimed to assist T2D patients. On the other hand, glucometers locally available in Colombia usually lack any USB or Bluetooth interface; this is the case of the Glucoquick G30a [9] and the OKmeter Match II [10]. These glucometers are regularly handed out to T1D and T2D patients by the Health Service Enterprises (known in Colombia as EPS). Therefore, if we aim to deploy an eHealth system for diabetic patients in the country; an alternative method for uploading glucometric data must be developed and tested; this is the core problem addressed in this report.

The manuscript is organized as follows: Section II discusses about the State of the Art in automated digitizing of glucometer data. Section III discusses about the System architecture for uploading glucometric data using a character recognition system based on AI techniques. Experimental tests and results are reported in Section IV, followed by conclusions in Section V.

II. State of the Art

A. System Requirements and Limitations of Traditional Glucometers

Based on the analysis of the technical challenges and limitations described in the recent literature, Table I describes critical requirements are established for a successful solution to digitize glucometer readings.

TABLE I. SYSTEM REQUIREMENTS SPECIFICATION

Requirement	*Specification*	*Refs*
R1. Robust Image Preprocessing	The system must preprocess images to manage variability due to lighting, screen glare, angle distortions, and resolution to enhance OCR accuracy.	[11], [12]
R2. Accurate Character Recognition (OCR)	The OCR model must achieve high clinical precision, mitigating effects of noise, artifacts, and low-quality captures.	[12], [13]
R3. Lightweight Architecture for Mobile Deployment	The system should function efficiently on smartphones with limited computational resources, ensuring accessibility for rural users.	[11], [14]
R4. User-Friendly Interaction	Interfaces must be intuitive, minimizing technical requirements for end-users, especially elderly patients or those with limited technological literacy.	[15]

According to Wyawahare et al. [16], most traditional glucometers still require manual data recording, limiting real-time decision-making capabilities. This manual process increases the risk of errors and decreases the efficiency of diabetes management, particularly in continuous monitoring scenarios. Furthermore, the lack of integration with health platforms constrains modern eHealth efforts.

B. Application of OCR and Machine Learning (ML) in Glucometer Digitalization

Tang et al. [17] and Tonyushkina and Nichols [13] have demonstrated that OCR combined with CNN-based ML architectures can extract accurate information from images captured directly from glucometer displays. However, this approach demands robust preprocessing strategies to normalize contrast, eliminate noise, and correct perspective distortions. These techniques are crucial to achieving R1 (Robust Image Preprocessing) and R2 (Accurate OCR Recognition).

Wolff et al. [11] and Khadem et al. [12] have identified that variability in image quality due to uncontrolled environmental conditions is a primary challenge. These findings reinforce the need for preprocessing solutions that handle diverse lighting and screen conditions.

C. Technical Challenges and Real-World Application

Khadem et al. [12] and Koca et al. [14] noted that small datasets often lead to overfitting, necessitating lightweight and generalizable models for mobile deployment — directly supporting R3 (Lightweight Architecture). Furthermore, integration difficulties with existing medical platforms (due to closed protocols) were reported by Wolff et al. [11], suggesting the need for standardization if the extracted data is to be interoperable within broader eHealth ecosystems.

Ben Ali et al. [18] and Hidalgo et al. [19] emphasized the necessity for clinical-grade validation of AI models in medical applications. These insights directly underline the need for reliable, highly accurate OCR systems to ensure clinical adoption and trust. Finally, Karter et al. [15] explores the effectiveness of self-monitoring blood glucose levels and glycemic control, which could support discussions around user interaction and system usability.

This work pointed out that user interface design plays a crucial role in technology acceptance, particularly among older adults and users with limited technical backgrounds, highlighting the importance of R4 (User-Friendly Interaction).

III. General System Architecture

The system for automated digitizing of glucometer data is encompassed within a larger eHealth system currently under development. Weekly feedback is provided by a specialist doctor based upon historical glucose data and patient medical history; the feedback usually comprises fine adjustments of insulin dosage and/or instructions on new medication. The system architecture for automated digitizing of glucometer data is structured into six functional stages: input, preprocessing, feature extraction, pattern recognition, classification, and output. This hierarchical organization ensures traceability from image acquisition to the final glucose value output, facilitating both technical implementation and potential future enhancements.

The modular design also supports scalability and interoperability with other telemedicine systems, particularly in rural or resource-limited settings. Figure 1 summarizes the operation of the eHealth system that requires daily glucose monitoring, followed by data upload to the platform.

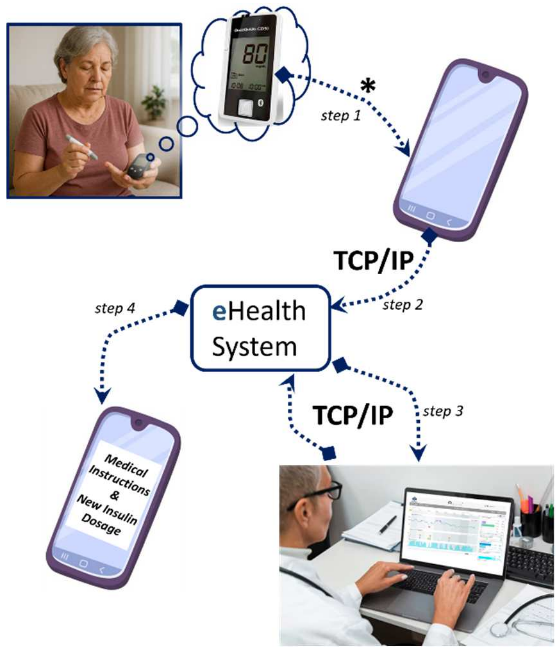

Fig. 1. Architecture of the eHealth System under development. Glucometer readout marked with * can be either stablished via Bluetooth, USB or photo capture and neural network analysis.

The aforementioned six stages of the system discussed are grouped and discussed in the detail in the following steps:

A. Image Capture and Segmentation of the Glucometer Screen

The recognition process begins with the acquisition of an image of the glucometer under controlled conditions. Image segmentation is then applied to isolate the specific region corresponding to the device's screen. This segmented area contains the relevant information—namely the glucose value, date, and time—which will be extracted in subsequent stages using detection and recognition techniques. A sketch is presented in Fig. 2 detailing the sequence of image capture, detection, and segmentation of the glucometer screen.

Fig. 2. Sequence of steps for image capture, detection and segmentation.

B. Image Recognition Algorithm-Generation of matrices in CSV format

To generate the input matrices for the Hopfield neural network, a preprocessing step was applied to images captured from a glucometer display. Using Python, the images were converted to grayscale and subsequently threshold to produce binary matrices, where each pixel is assigned a value of 1 or 0 based on a fixed threshold.

The binarized image was visualized using Matplotlib and exported to a CSV file for reproducibility and further analysis. The resulting binary matrix was then flattened into a one-dimensional vector and transformed into the bipolar format required by the Hopfield model (mapping 0 to 1 and 1 to -1). This preprocessing step ensures that the input data is appropriately formatted for pattern recognition tasks within the network. This whole process is described in Section IV, including Python functions that are listed in each class for reproducibility purposes.

C. Training with Hopfield Neural Network

A Hopfield Neural Network (HNN) is required for the recognition of numerical patterns represented as binary matrices. Each digit is encoded into a matrix form, simulating a simplified image, which serves as input to the network. Through the use of associative memory and Hebbian learning, the network is trained to store and recall these numerical patterns, enabling it to recognize and correct distorted or incomplete inputs. This approach demonstrates the ability of Hopfield NN to perform robust pattern recognition tasks with high tolerance to noise, making them suitable for applications involving low-resolution or degraded visual data.

D. The Convolutional Neural Network

A Convolutional Neural Network (CNN) model can be used to assist in the accurate extraction of digits from real-world images, enhancing the system's robustness against variations in quality or lighting conditions. We designed a modular, object-oriented architecture to automatically recognize digits displayed on glucometer screens using CNNs. The system consists of several classes, each with a specific function: acquiring and preprocessing the image, segmenting the digits, and finally classifying them with the neural network. The workflow begins by capturing or loading the image, which is then processed and segmented to extract each individual digit. The system uses a trained CNN to recognize these digits and compiles the results along with a confidence score for each prediction. This modular structure not only makes it easier to optimize and maintain each part of the system, but also allows us to adapt and scale the solution for different devices or image conditions, providing a robust and flexible tool for automatic digit recognition in medical devices.

IV. Preliminary Results

This section presents the system design and implementation using a set of structured *Unified Modeling Language (UML)* diagrams—complemented by pseudocode—that collectively illustrate the architecture, behavior, and component interactions of the digit recognition process. These visual and algorithmic representations are directly aligned with the Python source code implementation, enhancing clarity, modular understanding, and traceability of the software design choices.

A. General implementations structure

As previously mentioned, this study is focused on providing a compatible solution for rural communities that lacks glucometers with USB/Bluetooth interface. Therefore, the system architecture next described can be viewed as a small step towards the development of the larger eHealth system. This whole process is summarized in Fig. 3 and Fig. 4 based on UML class diagrams.

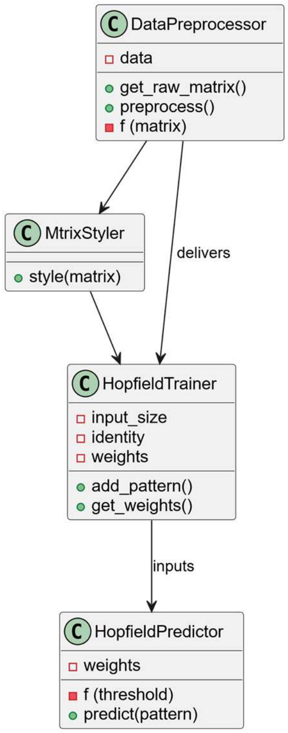

Fig. 3. UML class diagram of Hopfield NN training components.

A Python-based implementation processes binarized image matrices representing digits, trains the Hopfield HNN with predefined patterns, and performs classification based on energy minimization dynamics.

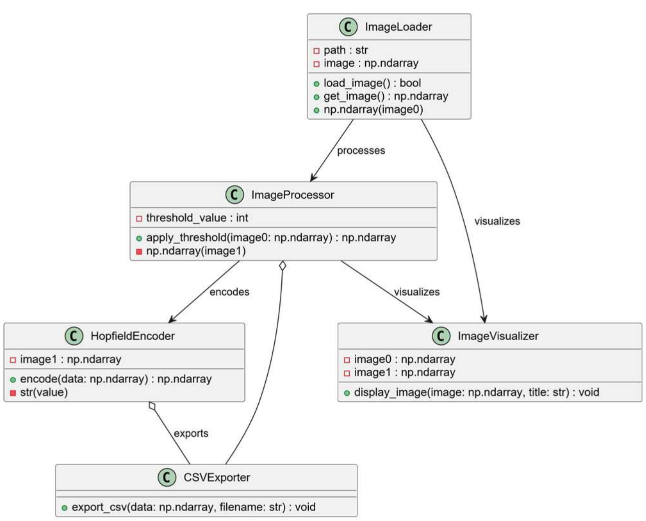

Fig. 4. General Class Diagram of the Modular System for OCR-Based Digit Recognition Using Hopfield Network.

To provide a detailed view of the training logic, Fig. 4 depicts the structure of the HNN training components. The *HopfieldTrainer* receives a series of digit patterns, flattens and normalizes them using bipolar transformation (2*X - 1), and computes their weight matrices through Hebbian learning. The final weight matrix is obtained by summing all the individual contributions. The system consists of modular entities such as *ImageLoader*, *ImageProcessor*, *HopfieldInputFormatter*, and *HopfieldPredictor*, each responsible for a specific stage in the digit recognition (cp. Fig. 4). Associations and compositions are shown to highlight the object-oriented relationships that can be derived from the current model implementation, as follows:

- *ImageLoader* loads digit matrices from .csv files.
- *ImageProcessor* resizes and preprocesses images.
- *HopfieldInputFormatter* converts matrices into binary input vectors for the network.
- *HopfieldPredictor* uses trained weights to classify unknown digits.

B. Interactions into the behavior of the system

The sequential stages are summarized in Fig. 5 for a better comprehension and each block represents a key component of the processing system.

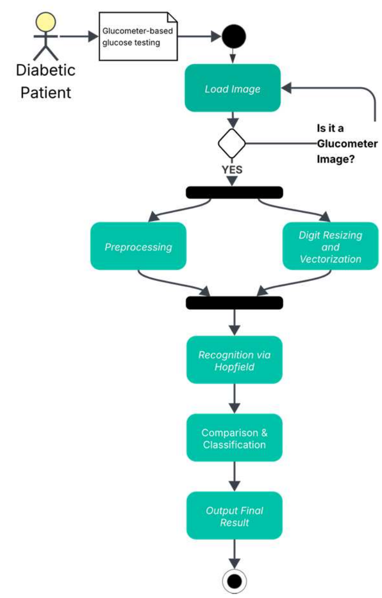

Fig. 5. UML activity diagram representing the recognition flow using Hopfield NN.

Fig. 7 illustrates the complete flow of the digit recognition process. It begins with the loading of an input matrix representing a test digit (e.g., from an image of the glucometer), continues through flattening and bipolar transformation, and concludes with classification using the trained weight matrix. An alternative path is also considered in case the output pattern does not converge, allowing for further preprocessing or rejection. To unify the visual representations with the algorithmic logic, *Algorithm 1* from Fig. 6 provides the high-level pseudocode of the recognition process. Each step in the pseudocode maps directly to specific blocks of the original code, including matrix transformation, Hopfield dynamics, and final classification.

Algorithm 1: Digit Recognition Using Hopfield Network

```
Input: Glucometer image
Output: Recognized numerical value
Step 1: Load glucometer image
Step 2: Preprocess image:
  2.1 Convert image to grayscale
  2.2 Apply adaptive thresholding (binarization)
  2.3 Remove noise using morphological filters
  2.4 Segment individual digit regions
Step 3: foreach segmented digit do
    3.1 Resize digit image to standard dimensions
    3.2 Convert resized image to binary input vector
    3.3 Recognition applying Hopfield Network:
      3.3.1 Initialize network with pre-trained patterns
      3.3.2 Run network dynamics until convergence
      3.3.3 Obtain and decode the output pattern
    3.4 Compare output with known digit templates
    3.5 Store recognized digit
end
Step 4: Combine recognized digits into final number
Step 5: Display or store the final result
```

Fig. 6. Pseudocode for digit recognition using Hopfield NN.

C. *Relationship between different components in a system*

Figure 7 presents the UML Component Diagram representing the functional decomposition of the digit recognition system designed for glucometer-based glucose readings. This high-level structural view organizes the software system into discrete components and interfaces that reflect the pipeline implemented in the original Python code.

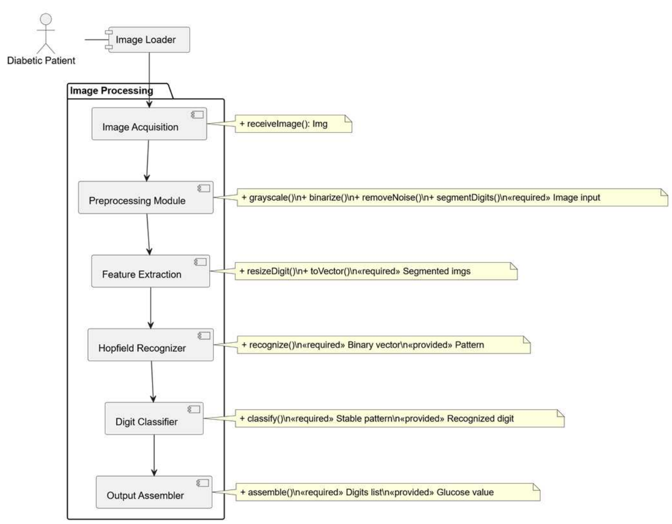

Fig. 7. UML component diagram of the Hopfield-based digit recognition system from glucometer images.

The process begins with the ***Image Acquisition*** component, triggered by the diabetic patient via an image loader. The acquired image is then passed to the ***Preprocessing Module***, which includes operations such as grayscale conversion, adaptive binarization, noise filtering, and digit segmentation. These steps correspond directly to the early preprocessing stages in the code where pandas and *numpy* are used to manipulate matrix representations of the digits. Next, the ***Feature Extraction*** component handles resizing and vectorizing segmented digits—critical steps to ensure compatibility with the Hopfield input format. The ***Hopfield Recognizer*** component manages the recall mechanism of the neural network, using binary input vectors and matching them against trained patterns via Hebbian-based weight matrices. This corresponds to the weight training and classification logic embedded in the original script using the dot product and bipolar transformation (2X - 1). Following this, the ***Digit Classifier*** evaluates the network's output and determines the corresponding recognized digit. Lastly, the ***Output Assembler*** combines all recognized digits into the final glucose reading, representing the system's output.

D. *Requirements fulfillment by the implementations system*

The implemented system, which integrates image preprocessing, matrix formatting, Hopfield Neural Network training, and digit classification, meets the specified design requirements (see R1-R2 from Table I), as follows:

R1. Robust Image Preprocessing. The system includes multiple preprocessing stages—grayscale conversion, adaptive thresholding, morphological filtering, and segmentation—executed through the *ImageProcessor* and *Preprocessing Module components*. These steps mitigate issues caused by lighting conditions, screen glare, and image distortion. The modular design ensures that each image is standardized before classification, directly enhancing OCR performance.

R2. Accurate Character Recognition (OCR). The use of a Hopfield Neural Network trained with representative digit patterns allows for robust digit recognition even when input patterns are noisy or partially distorted. The bipolar transformation (2·X – 1) and energy-based convergence mechanism implemented in the code help match test patterns to stored templates, satisfying the requirement for high recognition accuracy.

R3. Lightweight Architecture for Mobile Deployment. The system relies on matrix operations and a simple associative memory model (Hopfield NN), avoiding complex deep learning models. This makes it well-suited for execution on low-resource devices such as local PCs. Even, no GPU acceleration is needed because it will be able by a cloud service, reducing computational overhead and supporting deployment in mobile or embedded environments.

R4. User-Friendly Interaction. The activity diagram and decision nodes illustrate user-facing flows, including validation of input images and prompts to reload if the image is invalid. This iterative feedback loop ensures that non-technical users can interact with the system intuitively, increasing its accessibility for diabetic patients.

E. *Quantity result from the Hopfield NN:*

In context of evaluating this classification model, the confusion matrix serves as a fundamental tool for visualizing and quantifying a model's performance. By comparing the predicted labels with the true labels for each class, the confusion matrix provides detailed insight into the types and frequencies of correct and incorrect predictions. This allows us to identify not only the overall accuracy of the model, but also specific patterns of misclassification that may reveal underlying challenges in the data or model architecture. The confusion matrix from Fig. 8 shows that our CNN classifies the evaluated digit classes with high accuracy.

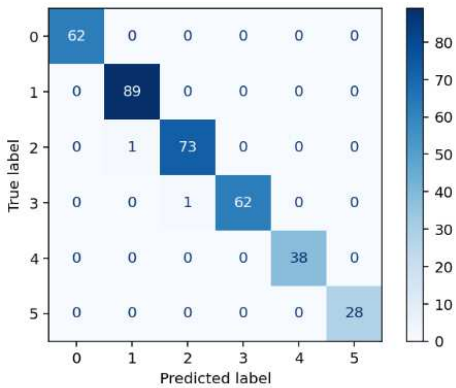

Fig. 8. HNN Confusion matrix results.

Most predictions match the true labels, as indicated by the strong concentration of values along the diagonal. We observe only a few misclassifications, where the model occasionally confuses visually similar digits. These results demonstrate that the HNN effectively learns and distinguishes the relevant features for digit recognition, delivering robust performance with minimal error. This high level of accuracy makes the model well-suited for practical applications in automatic digit recognition, such as those required for medical device interfaces. In addition, the low number of off-diagonal values highlights the model's ability to generalize to new data and avoid systematic confusion between specific digit classes. The rare errors likely result from similarities in digit appearance or noise in the input images, rather than from limitations in the model's design.

V. Conclusion

In this study, the combination of CNN and HNN neuronal networks let us assemble a digit recognition system for commercially available glucometers on the basis of images. The system outputs meaningful data for patient's glucometric data, date and time of measurement. Then, the proposed system is designed to recognize digits captured from glucometer displays using an HNN. We present the UML diagrams depicting the primary system and Hopfield NN training components. We use the confusion matrix to assess the effectiveness of our CNN in recognizing digits from medical device displays, offering a clear and interpretable summary of the model's strengths and areas for improvement. Overall, the confusion matrix confirms that our HNN-based approach reliably and effectively recognizes digits, supporting its use in real-world healthcare settings where accuracy and consistency are essential. The system was successfully tested on Glucoquick glucometers, which are popular in rural areas of Colombia, under conditions of daylight and artificial lighting.

References

[1] H. Vargas-Uricoechea and L. Á. Casas-Figueroa, "An Epidemiologic Analysis of Diabetes in Colombia," *Ann Glob Health*, vol. 81, 2015.

[2] V. Mohan, R. Unnikrishnan, S. Shobana, M. Malavika, R. M. Anjana, and V. Sudha, "Are excess carbohydrates the main link to diabetes & its complications in Asians?," *Indian J Med Res*, vol. 148, no. 5, 2018.

[3] B. J. Philip, M. Abdelrazek, S. Barnett, A. Bonti, and J. Grundy, "Toward a Unified mHealth Platform: A Survey of Current User Challenges and Expectations," *IEEE Access*, pp. 19876–19891, 2023.

[4] F. León-Vargas *et al.*, "Is a cloud-based platform useful for diabetes management in Colombia? The Tidepool experience," *Comput Methods Programs Biomed*, vol. 208, p. 106205, Sep. 2021.

[5] S. S. Crossen, C. C. Romero, C. Lewis, and N. S. Glaser, "Remote glucose monitoring is feasible for patients and providers using a commercially available population health platform," *Front Endocrinol (Lausanne)*, vol. 14, Feb. 2023, doi: 10.3389/FENDO.2023.1063290.

[6] M. Ng, E. Borst, A. Garrity, E. Hirschfeld, and J. Lee, "Evolution of Do-It-Yourself Remote Monitoring Technology for Type 1 Diabetes," *J Diabetes Sci Technol*, vol. 14, no. 5, pp. 854–859, Sep. 2020.

[7] Roche, "Medidor de glucemia Accu-Chek Aviva | Asistencia Accu-Chek," Accessed on April 1, 2025. Accessed: Mar. 31, 2025. [Online]. Available: https://support.rochediabetes.com/es/aviva-overview

[8] I. LifeScan, "Glucose meter - OneTouch® Ultra® 2 meter | OneTouch Solutions," Accessed on April 1, 2025. Accessed: Mar. 31, 2025. [Online]. Available: https://shop.onetouch.com/onetouch-sup-reg-sup-ultra-sup-reg-sup-2-meter/product/OTSUS05_0019?srsltid=AfmBOopmQU9CPRjoAG9X52ek1_e9UF2VUPIMgQ3pee72iryObOFyQldF

[9] Diabetrics, "Glucómetro G30a esencial para el control de diabetes," Accessed on April 1, 2025. Accessed: Mar. 31, 2025. [Online]. Available: https://www.diabetrics.com/g30a

[10] Okbiotech, "OKmeter Match II Sistema de control de glucosa en sangre, medidor de glucosa en sangre - OK Biotech Co., Ltd.," Accessed on April 1, 2025. Accessed: Mar. 31, 2025. [Online]. Available: https://www.okbiotech.com/es/product-372294/Sistema-de-control-de-la-glucemia-okmeter-Partido-II-medidor-de-glucosa-sangu%C3%ADnea.html

[11] M. K. Wolff, H. G. Schaathun, A. L. Fougner, M. Steinert, and R. Volden, "Mobile Software Development Kit for Real Time Multivariate Blood Glucose Prediction," *IEEE Access*, vol. 12, 2024.

[12] H. Khadem, H. Nemat, J. Elliott, and M. Benaissa, "In Vitro Glucose Measurement from NIR and MIR Spectroscopy: Comprehensive Benchmark of Machine Learning and Filtering Chemometrics," *Heliyon*, vol. 10, no. 10, p. e30981, May 2024.

[13] K. Tonyushkina and J. H. Nichols, "Glucose Meters: A Review of Technical Challenges to Obtaining Accurate Results," *J Diabetes Sci Technol*, vol. 3, no. 4, pp. 971–980, Jul. 2009.

[14] Ö. A. Koca, H. Ö. Kabak, and V. Kılıç, "Attention-based multilayer GRU decoder for on-site glucose prediction on smartphone," *J Supercomput*, vol. 80, no. 17, pp. 25616–25639, Nov. 2024.

[15] A. J. Karter *et al.*, "Self-monitoring of blood glucose levels and glycemic control: the Northern California Kaiser Permanente Diabetes registry∗," *Am J Med*, vol. 111, no. 1, pp. 1–9, Jul. 2001.

[16] M. Wyawahare, A. Aadarsh, and U. Anuse, "Market Analysis of Various Types of Rapid Testing Glucose Level Equipments," in *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, IEEE, Feb. 2024, pp. 311–316.

[17] L. Tang, S. J. Chang, C.-J. Chen, and J.-T. Liu, "Non-Invasive Blood Glucose Monitoring Technology: A Review," *Sensors*, vol. 20. 2020.

[18] J. Ben Ali, T. Hamdi, N. Fnaiech, V. Di Costanzo, F. Fnaiech, and J.-M. Ginoux, "Continuous blood glucose level prediction of Type 1 Diabetes based on Artificial Neural Network," *Biocybern Biomed Eng*, vol. 38, no. 4, pp. 828–840, 2018.

[19] J. I. Hidalgo, J. M. Colmenar, G. Kronberger, S. M. Winkler, O. Garnica, and J. Lanchares, "Data Based Prediction of Blood Glucose Concentrations Using Evolutionary Methods," *J Med Syst*, 2017.