G. Bekaroo et al. (Eds.)

doi:10.3233/AISE230019

Cloud-Based Monitoring System for Personalized Home Medication

Ahsan ISMAIL, Mario FIORINO, Musarat ABBAS, Madiha Haider SYED and Zaib ULLAH

Abstract. This paper introduces an Artificial Intelligence (AI)-enabled system to assist patients to follow a treatment plan at home. The deep learning model is a Convolutional Neural Network (CNN) classifier that is able to detect a drug even when shown in different orientations. The CNN model is trained for each patient based on his/her prescription medicine schedule. The advantage of the system is the dynamic functionality that makes it a good solution for personalized medication. The GUI demonstrates that the system can assist patients in taking the correct drug and prevent medication errors.

Keywords. Artificial intelligence, reinforcement learning, deep learning, medical treatment, medication error

1. Introduction

The occurrence of diseases among individuals has been on the rise in recent times, leading to a rise in the use of prescribed medicines worldwide [5]. Home treatment often requires patients to complete a medication course[7], with many patients being prescribed two or more drugs. However, due to sensory disabilities such as visual or hearing impairment, or amnesia, patients may have difficulty completing the course successfully [13]. It is common for older patients to have sensory disabilities such as visual and hearing impairments, as noted in studies [2][23]. As per the report published by the World Health Organization (WHO), around 65% of people aged above 50 suffer from visual and hearing impairments. Due to these sensory disabilities or illiteracy, most patients end up taking the wrong dosage of medication during their home treatment [20,4]. Additionally, some patients tend to forget to take their medication on time due to amnesia. These medication errors and the resulting Adverse Drug Events (ADE) can have serious consequences[19]. ADEs are especially prevalent among older people [10]. WHO reports that 30% to 50% of patients who receive home treatment take the wrong dosage, which is responsible for most deaths. In the United States, medication errors affect over 7 million patients every year and are the eighth leading cause of death [27]. While general pill reminder systems have been developed to remind patients of their medication schedule, they have not addressed the issue of medication errors caused by sensory-disabled patients.

Recently, with the development in machine learning algorithms [18] and neural network [16], the aim of Artificial Intelligence (AI) has become a step closer.

AI has significant application in many areas including:healthcare [8,25], UAVs, 5G and autonomous control [15], risk management [20,26], communication [14,11].

The solution to this problem proposed in [3], given the definition of CNN Based prediction model to recognize medicines and assist patients with sensory disabilities to take the correct medicine. However till now not any cloud-based monitoring systems have been developed that assist multiple patients by providing personalized monitoring assistance to them. As in recent development, the deployment of ML and deep learning models in cloud computing applications increasing rapidly. So our contribution to extending the idea given in [3] to cloud computing. Cloud computing has a multi-tenancy feature, which means that multiple clients can use the same computing resources without being aware of each other, and each client's data is not accessible to other clients. Our proposed solution to the problem of medication errors caused by sensory disabilities in patients is a cloud-based CNN monitoring system that provides personalized assistance to each patient during medication. For creating personalized assistance, we have to train a model for each user separately. Since each patient has different prescribed medicines for different diseases, it's impractical to build one global model that can provide personalized assistance. Therefore, we proposed that each user uploads a minimum of 20 images of each medicine prescribed to them for home treatment to the server to train a personalized CNN model. This approach has not been explored in the literature, and we believe it can significantly reduce medication errors.

We used a Convolutional Neural Network (CNN) to recognize the medicine taken by the patient. The rest of the paper is organized as follows: section 2 provides an overview of related work, while Section 3 presents the technical background of neural networks. Section 4 describes our methodology and experimentation, and section 5 provides detailed information about the experimental results and analysis. Finally, in section 6, we conclude the work.

2. Related Work

The Autoreminder System [22] uses partially observable Markov decision processes to plan and schedule the Nursebot system to assist in-home therapy. But the proposed architecture is not capable of preventing medication errors and is mainly designed only for the reminding process. The work of [9] employs smartphones that identify drugs by quantifying properties like color, size, and shape. However, for accurate estimation, such a methodology requires a marker to be used with known dimensions. In the work of [17], an intelligent pill reminder system is presented that consists of a pill reminder component and a verification component, however, it is not a cloud-based solution.

A working method is developed in [24] on the usefulness of a smart home to assist patients with a treatment process. The system initiates when a new drug prescription is advised by the doctor. An electronic system produces a QR code that is delivered with the prescription, indicating the time period, visit details, and medication workflow information. The set of information is utilized by an expert system that manages all data produced by the prescription. The methodology assists the subjects with no cognitive disability. There is no customization of the solution depending on the patient's skills. A time series-based method is presented in [21] for the early identification of an increase in hypertension. The authors have designed a hybrid deep network by integrating a deep neural network, a CNN, and a long-term memory signal.

Implementation of different AI and IoT-based proposals for remote healthcare monitoring have been reviewed in [1]. Similarly, different health monitoring solutions are proposed in [16,12,6]. However, these studies are not for personalized use.

The proposed work uses a cloud-based CNN monitoring system that provides personalized assistance to each patient during medication.

3. Technical background

This section presents a brief introduction to neural networks and CNN.

3.1. Artificial Neural Networks

The artificial neural network simply called Neural network is defined as “a method in artificial intelligence that teaches computers to process Data in a way that is inspired by the human brain”. Neural networks are a type of machine learning algorithm that is designed to simulate the behavior of the human brain. They are composed of layers of interconnected nodes, called ”neurons,” which are organized into input, hidden, and output layers. A single neuron, also known as a perceptron, is the simplest unit in a neural network. It is a mathematical function that takes one or more inputs, applies weights to them, and produces a single output. The basic structure of a single neuron includes: 1. Inputs: These are the values that are fed into the neuron. They could be binary values, continuous values, or discrete values. 2. Weights: Each input is associated with a weight, which determines the strength of the input's influence on the neuron's output. The weights are learned during training and are adjusted to minimize the error between the neuron's output and the expected output. 3. Bias: The bias is a constant value that is added to the weighted sum of the inputs. It allows the neuron to learn patterns even when all the inputs are zero. 4. Activation function: The activation function takes the weighted sum of the inputs and bias and applies a non-linear transformation to produce the neuron's output. This nonlinearity is important because it allows the neuron to learn complex patterns and relationships in the data. The output of a single neuron can be used as input to other neurons in a neural network, forming layers and hierarchies of processing. Single neurons can be used for a variety of tasks, including binary classification, regression, and pattern recognition. However, they are limited in their ability to learn complex patterns and relationships in the data. To overcome these limitations, multiple neurons are combined to form deep neural networks, which can learn complex patterns and relationships in the data through hierarchical processing. The weights and biases in a neural network are learned through a process called backpropagation, which adjusts the weights and biases based on the error between the network's output and the desired output. This allows the network to learn to make more accurate predictions over time. There are many different types of neural networks, each with its own strengths and weaknesses. Some common types of neural networks include feedforward neural networks, recurrent neural networks, convolutional neural networks, and deep neural networks.

3.2. Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of neural network that is particularly well-suited for processing data that has a grid-like structure, such as images, video, or audio. The basic idea behind a CNN is to apply a series of filters, or "convolutions," to the input data in order to extract meaningful features. A typical CNN consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for extracting features from the input data, while the pooling layers are used to reduce the dimensionality of the data and help prevent overfitting. The fully connected layers are used to make the final prediction based on the features extracted by the convolutional layers.

During training, the weights and biases in the network are adjusted using a process called backpropagation, which calculates the gradient of the loss function with respect to the weights and biases and uses this to update them in the direction that minimizes the loss.

One of the key advantages of CNNs is their ability to learn spatial hierarchies of features. For example, the first convolutional layer might learn simple features like edges and corners, while later layers might learn more complex features like object parts and textures. This hierarchical approach to feature learning allows CNNs to achieve state-of-the-art performance on a wide range of computer vision tasks, including image classification, object detection, and semantic segmentation. Overall, convolutional neural networks are a powerful tool for processing grid-like data and have become an essential part of the modern deep learning toolbox.

3.3. Transfer Learning

Transfer learning is a machine learning technique where a pre-trained model is used as a starting point for a new task, rather than training a new model from scratch. The idea behind transfer learning is that the knowledge learned by the pre-trained model on a previous task can be reused and transferred to the new task, allowing the new model to achieve better performance with fewer data and training time. In transfer learning, the pre-trained model is typically a deep neural network that has been trained on a large dataset, such as ImageNet, to solve a particular task, such as image classification. The pre-trained model is then modified by adding new layers or replacing existing ones to adapt it to the new task. The new layers are typically added on top of the pre-trained layers, and only the weights of the new layers are trained using the new dataset.

Transfer learning can be used in a variety of scenarios, such as:

1. When the new dataset is too small to train a deep neural network from scratch.

2. When the new dataset is different from the original dataset used to train the pre-trained model, but has some similarities.

3. When the computational resources required to train a deep neural network from scratch are not available.

Transfer learning has been shown to be an effective technique for a wide range of tasks, including image classification, object detection, and natural language processing. By leveraging the knowledge learned by pre-trained models, transfer learning can help reduce the time and resources required to train deep neural networks, while also improving their performance on new tasks.

4. System Model

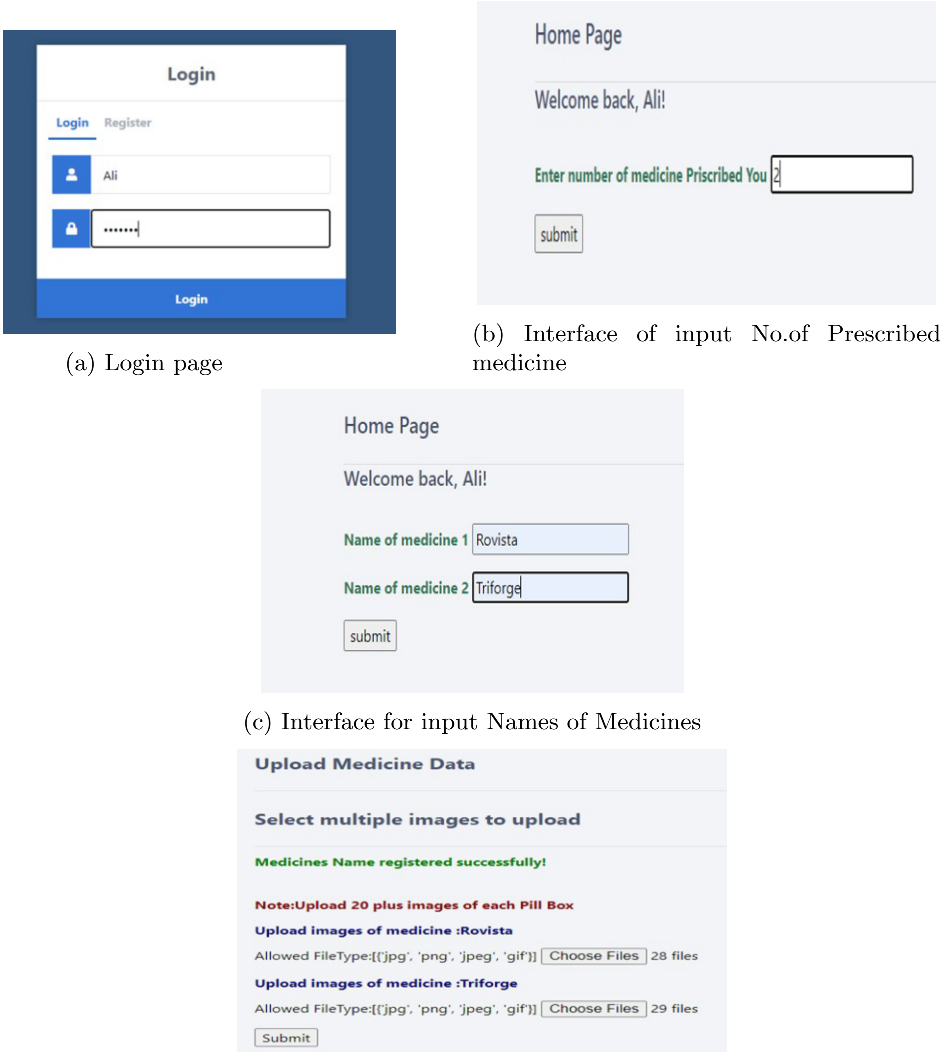
We have designed a web application using the Flask framework to set up a client-side interface for users. The application requires new users to register first and then log in using their account, as shown in (1a). Once the user logs in, an interface will appear as shown in (1b), where the user can input the number of prescribed medicines and submit it to access the interface where they can input the names of the medicines in their prescribed schedule, as shown in (1c). After submitting the names of the medicine, the user access the interface where they can upload medicine image data with 20 or more images of each medicine, as shown in (1d). The personalized model will be trained on the server side once the user uploads the data, and an admin will select which user model to train. Once the admin starts the model training, the model will be saved with the file name (username.h5) in .h5 format. The model will be available to users for making predictions once it is trained.

4.1. Data-collection

We manually captured images of pill boxes under different backgrounds and lighting conditions, as illustrated in Figure 2. We selected seven commonly used medicines in Pakistan, as shown in Figure 3. The collected data was then uploaded by the user from the client side. We utilized pre-processing and augmentation techniques to account for the small amount of data collected from the user side. Our pre-processing methods involved standard normalization, label collection, and resizing. In addition, we utilized various augmentation techniques, such as rotation, width shift, height shift, shear range, zoom range, horizontal flip, brightness range, and fill mode ('nearest'). These augmentation techniques improved the quality and quantity of the data.

4.2. Training Phase

After uploading the data to the server, the next step is to select a framework for building a deep learning model. There are several frameworks available, such as TensorFlow, Keras, PyTorch, etc., that come with vast libraries. We chose the Keras framework to build a CNN model on the admin side. Keras sequential API makes building a CNN model pretty simple, and it is open-source, fast experimentation with deep neural networks, user-friendly, and extensible. We decided



Login

Login Register

Ali

•••••••

Login

(a) Login page

Home Page

Welcome back, Ali!

Enter number of medicine Priscribed You 2

submit

(b) Interface of input No.of Prescribed medicine

Home Page

Welcome back, Ali!

Name of medicine 1 Rovista

Name of medicine 2 Triforge

submit

(c) Interface for input Names of Medicines

Upload Medicine Data

Select multiple images to upload

Medicines Name registered successfully!

Note:Upload 20 plus images of each Pill Box

Upload images of medicine :Rovista

Allowed FileType:[('jpg', 'png', 'jpeg', 'gif')] Choose Files 28 files

Upload images of medicine :Triforge

Allowed FileType:[('jpg', 'png', 'jpeg', 'gif')] Choose Files 29 files

Submit

(d) Upload images data

Figure 1. GUI of proposed system

to build a custom model instead of transfer learning because our dataset is entirely different from the datasets used in pre-trained models like VGG16, ResNet, EfficientNet models, etc.

We set up a custom model as shown in Figure 4. This model consists of 3 convolution layers with 32, 64, and 128 filters, respectively, each followed by max-pooling layers. Each convolution layer uses the Relu activation function. After the convolution layer, we added a Flatten layer and two dense layers with a Relu activation function. To avoid overfitting, we added a dropout of 0.5 after the first dense layer. The last dense layer uses softmax activation to output the probabilities for each class, as our dataset has seven classes. The input size for the

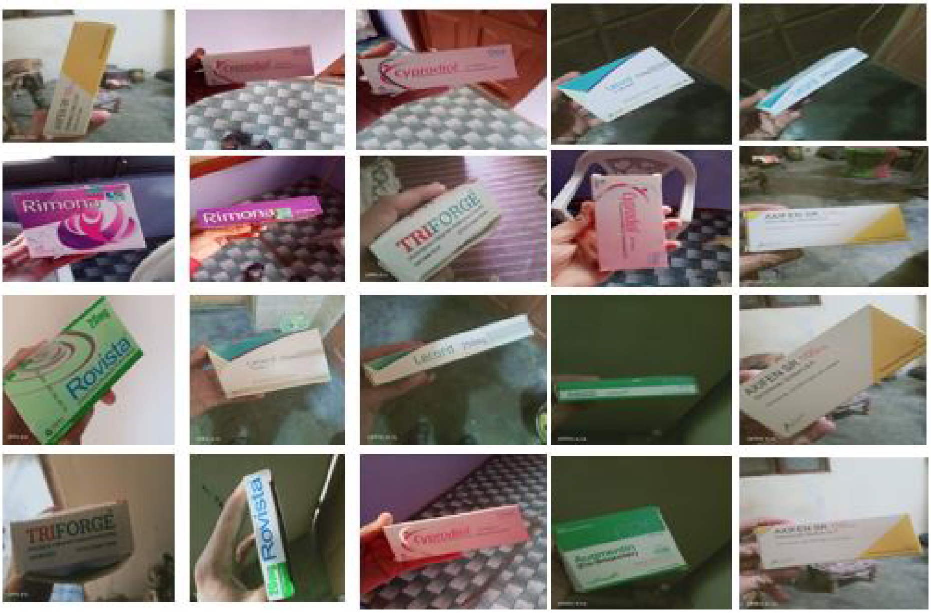
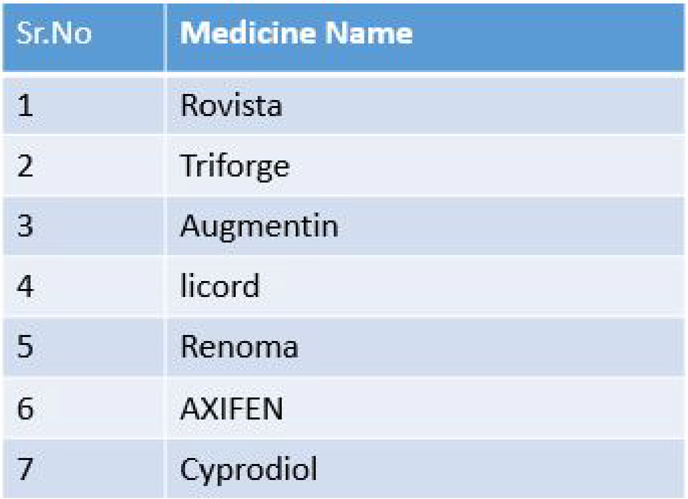


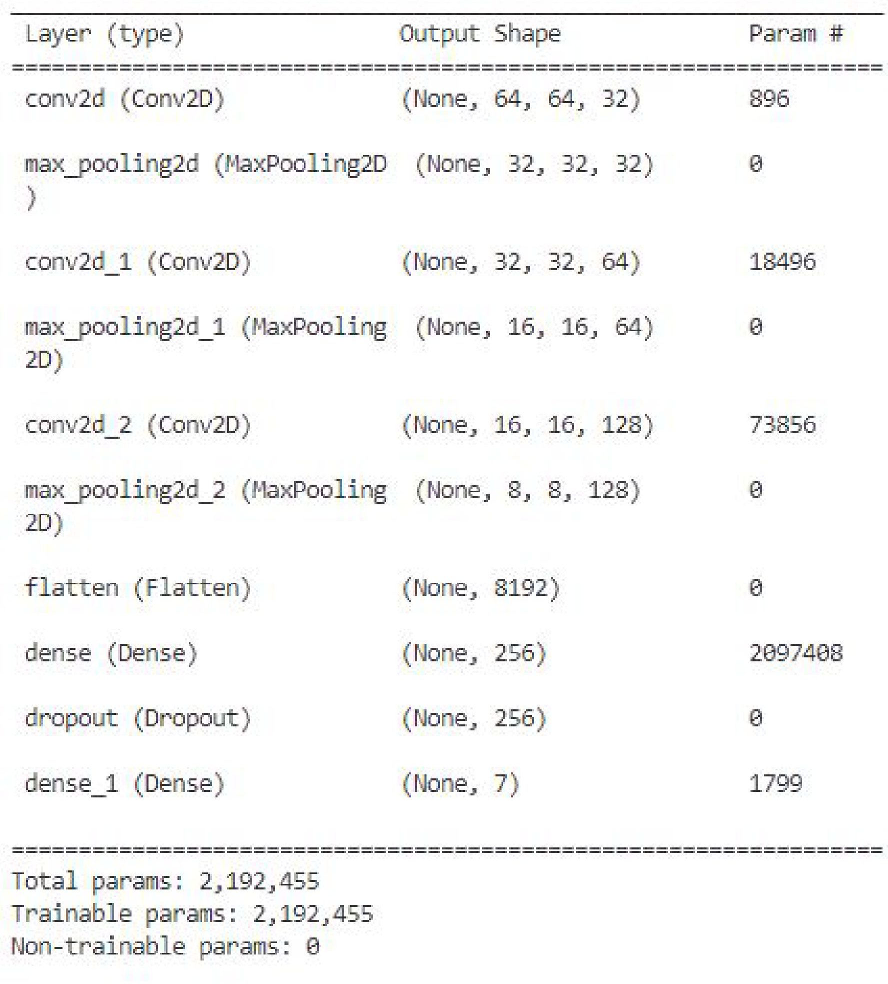
Figure 2. Data collection



Sr.No	Medicine Name
1	Rovista
2	Triforge
3	Augmentin
4	licord
5	Renoma
6	AXIFEN
7	Cyprodiol

Figure 3. Medicine names

first convolution layer is set to (64, 64, 3). We split our dataset into training data and validation with 15% of the data used for validation and 85% for training. Categorical cross-entropy was used as the loss function, and we applied a learning rate of 0.001 and a momentum of 0.9 to the Adam optimizer for optimization. Ultimately, the model is trained on both the training and validation datasets to improve its accuracy and prepare it for making predictions on new data.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 32)        896

 max_pooling2d (MaxPooling2D  (None, 32, 32, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 16, 16, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 8, 8, 128)        0
 2D)

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 256)               2097408

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 7)                 1799

=================================================================
Total params: 2,192,455
Trainable params: 2,192,455
Non-trainable params: 0
```

Figure 4. Model Summary

5. Results

After setting up the architecture of the custom CNN model, the model is evaluated using training and validation data. Once the evaluation is complete, the model is saved in the ”username.h5” format, which is mapped to the patient’s name and can be tested on the Flask web application from the client side for prediction. The model’s performance is measured using classification accuracy and loss function, which are displayed in Figure 5 and Figure 6, respectively.

The model achieved its best performance with a validation accuracy of 97% and a validation loss of 0.08. The model was trained using the user dataset on the admin side of the Flask web application. Once the model was trained, it was saved with a username mapping. The deployment of the model for prediction testing is demonstrated on the client-side application, and its GUI for uploading images is shown in Figure 7. Users can upload images for prediction from their side, as shown in Figure 8.

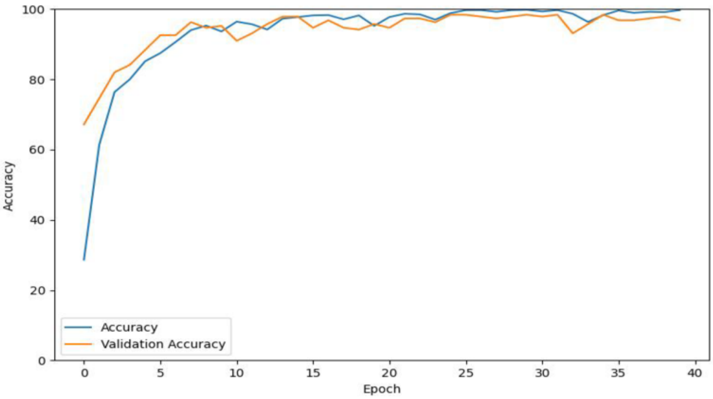


Figure 5. Performance in terms of accuracy

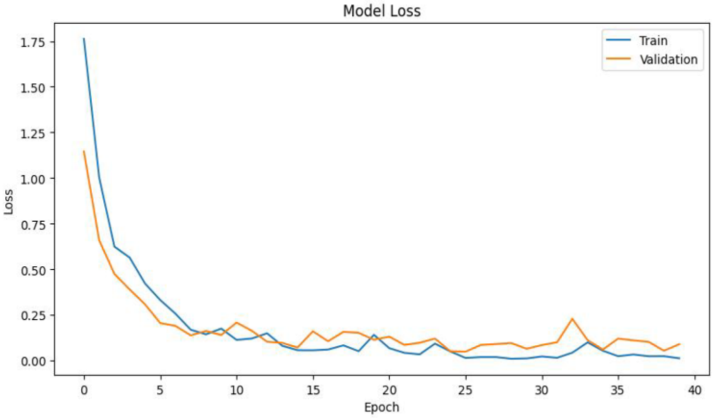


Figure 6. Performance in terms of loss function

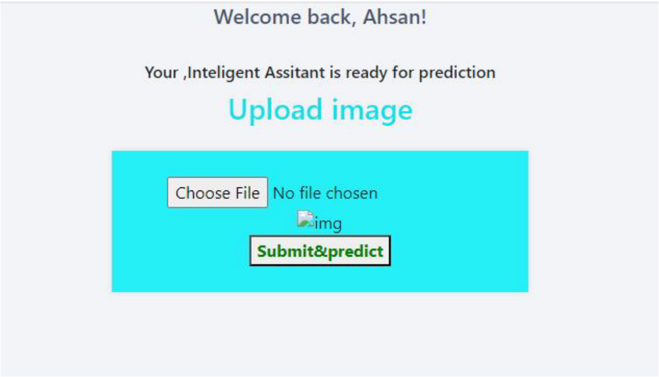
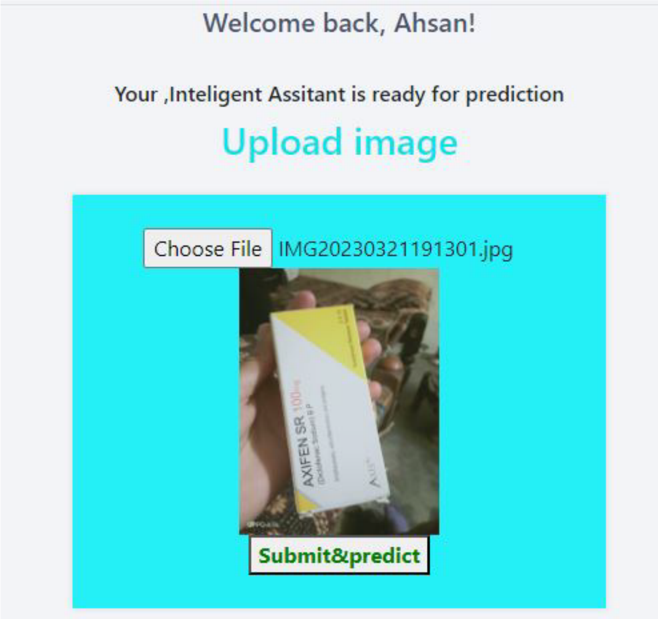


Figure 7. GUI-upload images interface



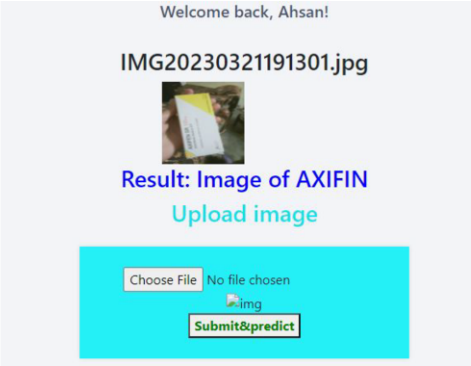
Welcome back, Ahsan!

Your ,Inteligent Assitant is ready for prediction

Upload image

IMG20230321191301.jpg

Figure 8. GUI-trained model ready for prediction



Welcome back, Ahsan!

IMG20230321191301.jpg

Result: Image of AXIFIN

Upload image

No file chosen

Figure 9. GUI- Prediction result

The Model predicts the image and shows the image label on the screen as the results shown in Figure 9.

6. Conclusion

We describe a personalized CNN monitoring system that operates as a pill reminder system in a cloud-based setup. A separate CNN model is trained for each patient based on their prescription medication data, with the goal of checking whether they are taking the correct medication. We developed a custom model setup that trains a personalized model on patients' data, with the trained model performing well on validation and prediction test data. This proposed work has the potential to improve patient adherence to medication.

References

[1] Mazin Alshamrani. Iot and artificial intelligence implementations for remote healthcare monitoring systems: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2021.

[2] Ko-Fang Chang, Kwang-Hwa Chang, Wen-Chou Chi, Shih-Wei Huang, Chia-Feng Yen, Hua-Fang Liao, Tsan-Hon Liou, Pin-Zhir Chao, and I-Chan Lin. Influence of visual impairment and hearing impairment on functional dependence status among people in taiwan—an evaluation using the whodas 2.0 score. *Journal of the Chinese Medical Association*, 81(4):376–382, 2018.

[3] Mario Ciampi, Antonio Coronato, Muddasar Naeem, and Stefano Silvestri. An intelligent environment for preventing medication errors in home treatment. *Expert Systems with Applications*, 193:116434, 2022.

[4] Antonio Coronato, Giuseppe De Pietro, and Giovanni Paragliola. A monitoring system enhanced by means of situation-awareness for cognitive impaired people. In *Proceedings of the 8th international conference on body area networks*, pages 124–127, 2013.

[5] Antonio Coronato, Giuseppe De Pietro, and Giovanna Sannino. Middleware services for pervasive monitoring elderly and ill people in smart environments. In *2010 Seventh International Conference on Information Technology: New Generations*, pages 810–815. IEEE, 2010.

[7] Antonio Coronato and Muddasar Naeem. Ambient intelligence for home medical treatment error prevention. In *2021 17th International Conference on Intelligent Environments (IE)*, pages 1–8. IEEE, 2021.

[8] Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964, 2020.

[9] Andreas Hartl. Computer-vision based pharmaceutical pill recognition on mobile phones. In *Proc. 14th Central European Seminar on Computer Graphics*, page 5. Citeseer, 2010.

[10] Kate L Lapane, Carmel M Hughes, Lori A Daiello, Kathleen A Cameron, and Janice Feinberg. Effect of a pharmacist-led multicomponent intervention focusing on the medication monitoring phase to prevent potential adverse drug events in nursing homes. *Journal of the American Geriatrics Society*, 59(7):1238–1245, 2011.

[11] Muddasar Naeem, Sajid Bashir, Muhammad Usman Khan, and Aqeel A Syed. Performance comparison of scheduling algorithms for mu-mimo systems. In *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 601–606. IEEE, 2016.

[13] Muddasar Naeem, Antonio Coronato, and Giovanni Paragliola. Adaptive treatment assisting system for patients using machine learning. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 460–465. IEEE, 2019.

[14] Muddasar Naeem, Antonio Coronato, Zaib Ullah, Sajid Bashir, and Giovanni Paragliola. Optimal user scheduling in multi antenna system using multi agent reinforcement learning. *Sensors*, 22(21):8278, 2022.

[15] Muddasar Naeem, Giuseppe De Pietro, and Antonio Coronato. Application of reinforcement learning and deep learning in multiple-input and multiple-output (mimo) systems. *Sensors*, 22(1):309, 2021.

[16] Muddasar Naeem, Giovanni Paragiola, Antonio Coronato, and Giuseppe De Pietro. A cnn based monitoring system to minimize medication errors during treatment process at home. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, pages 1–5, 2020.

[17] Muddasar Naeem, Giovanni Paragliola, and Antonio Coronato. A reinforcement learning and deep learning based intelligent system for the support of impaired patients in home treatment. *Expert Systems with Applications*, page 114285, 2020.

[18] Muddasar Naeem, Syed Tahir Hussain Rizvi, and Antonio Coronato. A gentle introduction to reinforcement learning and its application in different fields. *IEEE access*, 8:209320–209344, 2020.

[19] Jonathan R Nebeker, Paul Barach, and Matthew H Samore. Clarifying adverse drug events: a clinician's guide to terminology, documentation, and reporting. *Annals of internal medicine*, 140(10):795–801, 2004.

[20] Giovanni Paragliola and Antonio Coronato. A reinforcement-learning-based approach for the planning of safety strategies in aal applications. In *Intelligent Environments 2018*, pages 498–505. IOS Press, 2018.

[21] Giovanni Paragliola and Antonio Coronato. An hybrid ecg-based deep network for the early identification of high-risk to major cardiovascular events for hypertension patients. *Journal of Biomedical Informatics*, 113:103648, 2021.

[22] Martha E Pollack. Planning technology for intelligent cognitive orthotics. In *AIPS*, pages 322–332, 2002.

[23] Parminder Raina, Micheline Wong, and Helen Massfeller. The relationship between sensory impairment and functional independence among elderly. *BMC geriatrics*, 4(1):1–9, 2004.

[24] M Ramljak. Smart home medication reminder system. In *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5, 2017.

[25] Syed Ihtesham Hussain Shah, Antonio Coronato, Muddasar Naeem, and Giuseppe De Pietro. Learning and assessing optimal dynamic treatment regimes through cooperative imitation learning. *IEEE Access*, 10:78148–78158, 2022.

[26] Syed Ihtesham Hussain Shah, Muddasar Naeem, Giovanni Paragliola, Antonio Coronato, and Mykola Pechenizkiy. An ai-empowered infrastructure for risk prevention during medical examination. *Expert Systems with Applications*, page 120048, 2023.

[27] Brigitta van Ewijk. Medication error prevention: Improving patient health outcome. 2018.