A Machine Learning-based pH Color Recognition for Monitoring Chronic Kidney Disease

Michael Lum
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
mlum3@csu.fullerton.edu

Rong Jin
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
rong.jin@fullerton.edu

2023 IEEE World AI IoT Congress (AIIoT) | 979-8-3503-3761-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/AIIoT58121.2023.10174521

***Abstract*—Nearly 15% of the adult American population are affected by Chronic Kidney Disease (CKD) according to the Center for Disease Control and Prevention (CDC). According to the National Institutes of Health (NIH), there are five stages of CKD, with each stage reflecting a different level of kidney function. In its early stages, CKD can often be asymptomatic. As a result, less than 10% of individuals are aware of having stage 1-3 CKD. Therefore, CKD is described as a "silent killer" for them. One common method of detecting and monitoring CKD is through blood or urine tests that measure biomarkers such as blood urea nitrogen (BUN), creatinine, or protein levels. However, these tests require patients to visit clinics for sample collection, and wait for results and feedback from their doctors, which causes much inconvenience and stress to the patient. A less burdensome alternative for diagnosing CKD is to measure BUN levels through a urine or saliva test using pH test strips. Although pH test strips are widely available, accurate determination of a patient's pH level through analysis of strip images remains challenging. This empirical pilot study aims to explore the possibility and effectiveness of using supervised machine learning models, such as KNN, SVM, and Neural Network, to predict pH values by recognizing the RGB profile data of pH test strips. The model settings are thoroughly explained, and the accuracy of the machine learning-based pH color recognition demonstrates promise in detecting CKD. Future research may focus on optimizing the dataset and machine learning models.**

***Index Terms*—Machine Learning, pH Color Recognition, Chronic Kidney Disease Monitoring**

I. Introduction

Nearly 15% of the adult American population are affected by Chronic Kidney Disease (CKD) according to the Centers for Disease Control and Prevention (CDC) [1]. Individuals with CKD are at higher risk of early death and complications with heart disease and stroke. The primary causes of CKD are high blood pressure and diabetes. According to the National Institutes of Health (NIH), there are five stages of CKD, with each stage reflecting a different level of kidney function. Stage 1 is characterized by a slight decrease in normal kidney function, which may not cause noticeable symptoms. Then, CKD progresses from a mild to moderate decrease in kidney function in stage 2, to a moderate to a severe decrease in stage 3, and finally to a severe decrease in stage 4, with symptoms worsening and medical intervention becoming necessary to manage the condition. The final Stage 5 is described as kidney failure, requiring patients to undergo a kidney transplant, and they may experience symptoms such as fluid retention and nausea, according to the Mayo Clinic [2].

Individuals in the early stages of CKD can be asymptomatic. As a result, less than 10% of individuals are aware of having stage 1-3 CKD. Thus, CKD is described as a "silent killer" due to the lack of symptoms in most individuals. At present, one common method of detecting and monitoring CKD is through blood or urine tests that measure biomarkers such as blood urea nitrogen (BUN), creatinine, or protein levels. However, these tests require patients to visit clinics for sample collection, and wait for results and feedback from their doctors, which causes much inconvenience and stress to the patient. [4] discusses the challenges driving the need to research alternative technologies to measure urea levels. A potential convenient alternative proposed for diagnosing CKD is to measure BUN levels through a urine or saliva test using pH test strips at home. The patient can use a test strip reader equipped with machine learning to determine BUN levels. Additionally, studies such as [5] and [6] have shown the potential of pH indicators in detecting BUN levels. These studies focus on utilizing pH for urea determination.

In [5], an assessment was conducted to explore the potential of salvia as a means to measure BUN levels. The researchers employed filter paper infused with urease and phenol red indicator. Urease is an enzyme that decomposes BUN molecules into smaller chemical compounds. Phenol red is a pH indicator, meaning that it changes color depending on pH levels. In the study, the infused filter paper was saturated with saliva solution. The urease broke down the BUN in the saliva into its constituent parts, causing the solution's acidity to increase and the phenol red indicator to change color in response to the acidity level. As the acidity level corresponds to the BUN concentration, the color change also indicates the BUN levels in the saliva. A smartphone was used to capture an image of the filter paper, which was then analyzed for RGB profiling using a mobile application specifically designed for this purpose. The RGB profiling was found to be closely related to the specified BUN concentrations.

In [6], a similar approach was adopted where urease, saliva, and a pH indicator were used. However, instead of infused filter paper, a small well made through 3D printing was employed to contain a solution of urease and pH indicator.

This well was sandwiched between a light source that emitted yellow light only and a light sensor. Saliva samples and the solution were introduced into the well and allowed to stabilize to ensure that the pH indicator completed the color change. Once the color change was complete, the light was directed into the well. The colored solution absorbed some of the light, while the remaining light contacted the light sensor. Based on the light's intensity, the light sensor produced a current that could be used to determine the BUN concentration in the saliva.

The aforementioned mentioned applications demonstrate the potential use of pH for urea measurement, which may alleviate the inconvenience of patients having to visit clinics for BUN assessments by providing an alternative method of collecting blood and urine samples using pH indicators, as well as reducing wait times for test interpretation by clinic staff and physicians, there has been limited research on accurately determining a patient's pH level through the analysis of pH test strip images. Therefore, this empirical pilot study aims to explore the utilization of supervised machine learning models to determine pH values by recognizing the colors of the pH test strips. The evaluation parameters and findings are clearly outlined, and the precision of the machine learning methods offers encouraging results for detecting CKD. Further research can refine the models and incorporate BUN-level readings into a mobile app, which could facilitate easy monitoring of CKD at home.

The paper is organized as follows. Section II reviews the relevant research. The experimental details are outlined in Section III, and Section IV discusses the performance analysis and our findings. This study is concluded in Section V, including the possible future works.

II. Related Work

Previous studies have shown the potential of using machine learning for color recognition. According to [3], K-Nearest Neighbors (KNN) models have the potential to be used for color classification and can accurately classify colors such as white, black, orange, green, yellow, red, blue, and violet. In [8], the utilization of various machine learning models for color-based tooth recognition is explored. Furthermore, a study discussed in [7] was carried out to explore the potential of using the Convolutional Neural Network (CNN) for processing RGB-D images. Hence, we will investigate the application of machine learning in recognizing the colors of pH test strips.

III. Experimental Setup

A. Dataset

For this study, we obtained the dataset from Kaggle [10] which consists of RGB profile data for pH test strips at different pH levels. pH test strips change color in response to a specific pH level, and the RGB profile represents the color of the pH test strip for a particular pH level. The dataset includes 653 entries of RGB profile data, along with their corresponding pH levels.

B. Models

We assessed the performance of three different supervised machine learning models: k Nearest Neighbor (KNN), Support Vector Machine (SVM), and Neural Network (NN) for this study. Implementation details and the performance analysis of each model test case are presented in Section IV.

C. Performance Metrics

The performance evaluation of KNN and SVM models involves a combination of accuracy and Mean Absolute Error (MAE) metrics for each test case. Accuracy indicates how well the trained models can correctly identify the pH value, while the MAE measures the average absolute difference between the predicted pH and the actual pH value across all of the dataset.

The performance of NN models is presented in the form of accuracy vs. epochs graphs. These graphs depict the correlation between accuracy and epoch for both the training and test sets. By analyzing these graphs, we can understand how effectively the neural network is trained to predict pH values.

IV. Performance Analysis

We evaluated the performance of all three types of models using multiple test cases with varying attribute parameters. For each test case, we conducted three iterations, randomly assigning 10% of the dataset to a test set while using the remaining data for the training set. Thus, unique training and test sets were used for every iteration. To facilitate our study, we utilized the machine learning libraries scikit-learn and TensorFlow.

A. KNN

We performed four test cases for the KNN model, with each test case utilizing a different KNN model configuration that varied in terms of the number of neighbors and data normalization parameters. Data preprocessing was performed on the x-variable (RGB profile) using two normalization methods: the 'l2' normalization method via the *preprocessing.normalize* function and channel-wise normalization. The formula utilized for channel-wise normalization (i.e., test case 4) was derived from [8]. *ball_tree* algorithm was selected due to RGB profile data having multiple dimensions. A summary of these configurations is presented in Table I.

Table II shows the accuracy of the KNN models, which ranges from 66.67% to 83.33%. Among these models, Test Case 2 achieved the highest accuracy of 83.33%. However, the accuracy did not improve with data preprocessing through normalization, which might be due to random outlier placement in either the training or test set. This could be a possible explanation for the wide variation in accuracy.

B. SVM

We evaluated the SVM model using scikit-learn's *SVC* function in four test cases, each with different kernel and degree attributes. Table III provides a summary of these configurations. As discussed in Section IV-D, misclassification

TABLE I. Overview of KNN Test Case Configurations

Test Case	KNN Model
1	*KNeighborsClassifier*(n_neighbors= 5, weights = None, algorithm = ‘ball_tree’)
2	*KNeighborsClassifier*(n_neighbors= 10, weights = None, algorithm = ‘ball_tree’)
3	*KNeighborsClassifier*(n_neighbors= 10, weights = None, algorithm = ‘ball_tree’) , data preprocessing with *preprocessing.normalize*
4	*KNeighborsClassifier*(n_neighbors= 10, weights = None, algorithm = ‘ball_tree’) , data preprocessing using the formula: R = R/(R+B+G), G = G/(R+B+G), B = B/(R+B+G)

TABLE II. Evaluation Results of KNN Model Test Cases

Test Case	Metrics	Iteration 1	Iteration 2	Iteration 3
1	Accuracy	0.7273	0.7727	0.8182
	MAE	0.5303	0.3939	0.3030
2	Accuracy	0.7121	0.7273	0.8333
	MAE	0.4697	0.4394	0.2727
3	Accuracy	0.6818	0.6667	0.7576
	MAE	0.4545	0.5909	0.3939
4	Accuracy	0.7273	0.6667	0.6667
	MAE	0.5152	0.5758	0.4394

of pH values can occur due to overlapping RGB profiles for specific pH values. To limit allowable misclassification, a larger C value of *SVC* function is used. Therefore, we set C is 1000 in all test cases. As the minimal impact of data preprocessing on KNN models implies that the dataset could have many outliers. The influence of outliers on model performance may be greater than that of normalization. As a result, data preprocessing was not applied to the SVM test cases.

TABLE III. Overview of SVM Test Case Configurations

Test Case	SVM Model
1	*SVC*(C=1000, kernel = ‘rbf’)
2	*SVC*(C=1000, kernel = ‘poly’, degree = 3)
3	*SVC*(C=1000, kernel = ‘poly’, degree = 4)
4	*SVC*(C=1000, kernel = ‘sigmoid’)

TABLE IV. Evaluation Results of SVM Test Cases

Test Case	Metrics	Iteration 1	Iteration 2	Iteration 3
1	Accuracy	0.6061	0.8636	0.7727
	MAE	0.5758	0.2576	0.3485
2	Accuracy	0.6818	0.7576	0.6364
	MAE	0.5000	0.3333	0.6364
3	Accuracy	0.6970	0.6515	0.7121
	MAE	0.4545	0.6364	0.4091
4	Accuracy	0.0909	0.0606	0.0758
	MAE	2.8636	2.8182	3.2727

Table IV presents the accuracy of SVM models with different kernels. The results indicate that models with *rbf* or *poly* kernels achieved higher accuracy levels ranging from 60.61% to 86.36% (test cases 1-3). However, as discussed earlier in the KNN section, the variation in accuracy might be due to the presence of random outliers in the dataset. On the other hand, the *sigmoid* kernel did not fit the dataset, resulting in poor hyperplane definition and lower accuracy.

C. NN

We conducted three test cases for NN models, with each case evaluating variations of the baseline model illustrated [9] as shown in Fig. 1. For the purpose of experimental comparison, test case 2 removed a hidden layer in the neural network model. The removal of a hidden layer simplifies the model and reduces the risk of overfitting the data. Test case 2 aimed to evaluate whether the removal of a hidden layer affected the model’s performance compared to test case 1. The validation split was set at 20% for each NN model, and 2000 epochs were used. Table V summarizes all test cases.

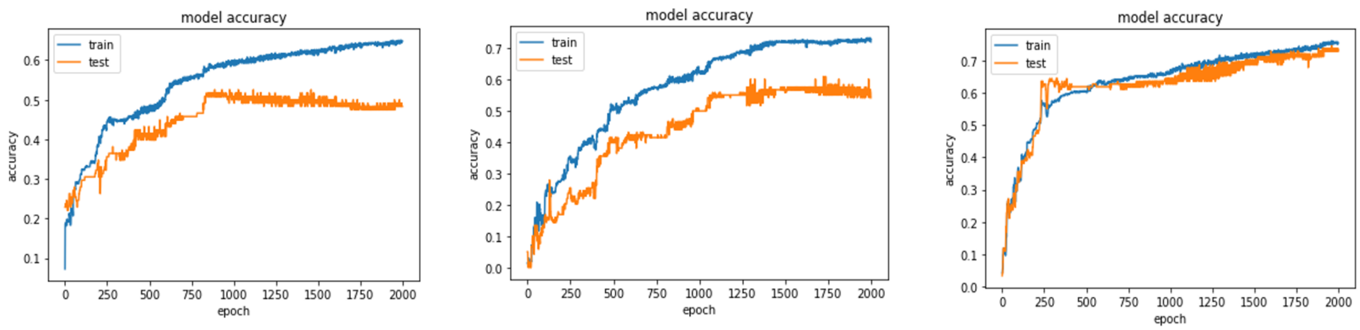
```
model = keras.Sequential([
    layers.Dense(3, kernel_regularizer=regularizers.l2(0.001), activation='relu',
    input_shape=[len(X_train.keys())]), #inputshape=[3]
    layers.Dense(24, kernel_regularizer=regularizers.l2(0.001), activation='relu'),
    layers.Dense(24, kernel_regularizer=regularizers.l2(0.001), activation='relu'),
    layers.Dense(16, kernel_regularizer=regularizers.l2(0.001), activation='relu'),
    layers.Dense(15)
  ])

optimizer = keras.optimizers.Adam(learning_rate=0.001)
loss_function = tf.keras.losses.CategoricalCrossentropy(from_logits=True)

model.compile(loss=loss_function,
                optimizer=optimizer,
                metrics=['accuracy'])
model.summary()
```

Fig. 1. Baseline Neural Network (NN) [9]

TABLE V. Overview of NN Test Case Configurations

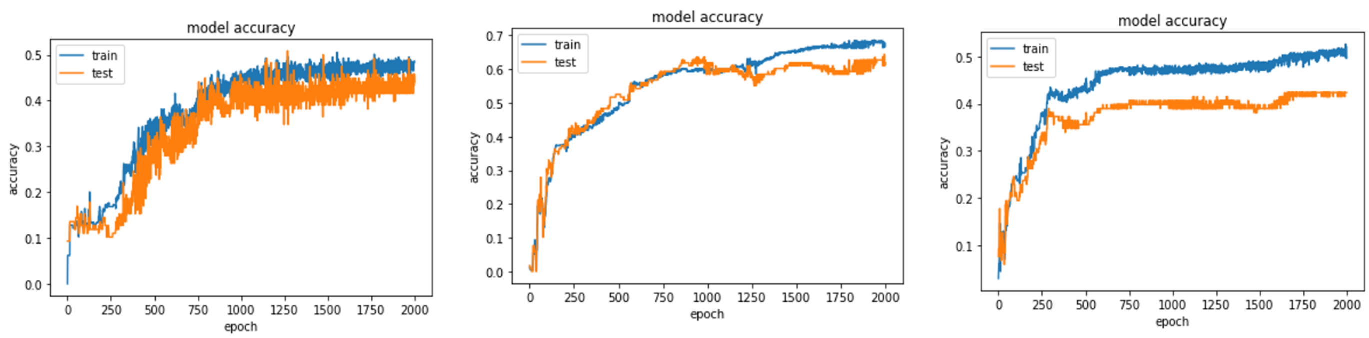
Test Case	NN Model
1	The baseline model shown in Fig. 1 using the ‘relu’ activation function in four hidden layers.
2	A variant of the baseline model using three hidden layers instead of four, with ‘relu’ activations applied to all three.
3	A variant of the baseline model where the activation function for all four hidden layers is changed from ‘relu’ to ‘sigmoid’.

The performance of the models is shown in Fig. 2, 3, and 4. The figures display the training and test set fits for each model. The neural network using the *relu* activation function in all layers consistently demonstrated the best performance with an average accuracy of over 70% for both training and test sets. However, removing a hidden layer (i.e., test case 2) resulted in decreased accuracy for the model in some cases, indicating that the accuracy may be affected by the partitioning of the data into training and validation sets. The performance of the NN models may also be affected by the presence of outliers in the dataset, similar to the KNN and SVM models. Additionally, as with the SVM models, the *sigmoid* activation function demonstrated poor performance for the selected dataset.



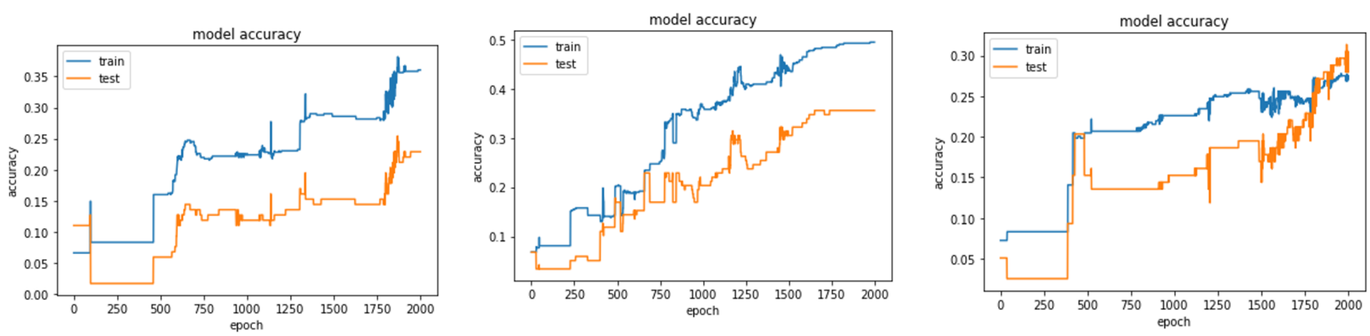
(a) The Accuracy of Iteration 1 (b) The Accuracy of Iteration 2 (c) The Accuracy of Iteration 3

Fig. 2. Evaluation results of Test Case 1 using the baseline NN model featuring four hidden layers with *relu* activations.



(a) The Accuracy of Iteration 1 (b) The Accuracy of Iteration 2 (c) The Accuracy of Iteration 3

Fig. 3. Evaluation results of Test Case 2 using a modified baseline NN model using three hidden layers with *relu* activations.



(a) The Accuracy of Iteration 1 (b) The Accuracy of Iteration 2 (c) The Accuracy of Iteration 3

Fig. 4. Evaluation results of Test Case 3 using a modified baseline NN model where all four hidden layer activations changed from *relu* to *sigmoid*.

D. Discussion

We suggest that defining more robust boundaries or distinct hyperplanes between RGB profiles for specific pH values could enhance the performance of machine learning models. The dataset available from Kaggle has few well-defined boundaries between RGB profiles for specific pH values, as shown in Fig. 5. Additionally, factors such as lighting, camera distance, camera focusing, and camera model can potentially affect the derivation of RGB profiles. One possible approach to obtaining a dataset with well-defined RGB boundaries is to control how pH test strip images are generated. [5] [6] have suggested the use of a fixture to collect data from pH indicators. The use of a fixture might possibly reduce the presence of outliers and improve the definition of hyperplanes between pH RGB profiles. This would aid in the creation of distinct hyperplanes and possibly improve machine learning model performance. Furthermore, the brand or model of pH test strips used to create the Kaggle dataset is not specified, which hinders the testing of machine learning models against physical images of pH test strips. With these factors, we created an artificial dataset using a colour chart (see Fig. 6) from the brand Hydrion's pH Plastic Test Strips 5.5-8.0 (part number 9700) in our study.

To create an artificial dataset, the RGB profile for each pH in the colour chart was gathered. 40 data points (RGB profiles) were created for each pH. This was done by ± 5 to the Red, Green, and Blue values for a given pH. This was repeated for every pH 40 times. This created a total of 480 RGB profiles that are used to train the model. A plot of the artificial dataset is shown in Fig. 7.

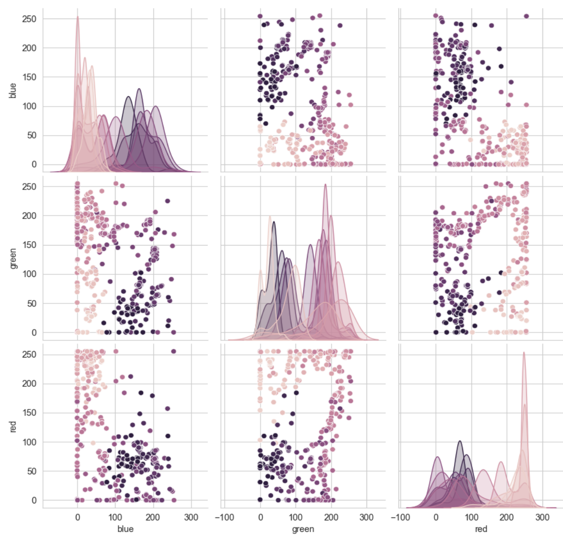


Fig. 5. Visualization of the Kaggle Dataset. It encompasses pH values ranging from 0 to 14. Each subplot's axes represent RGB values.

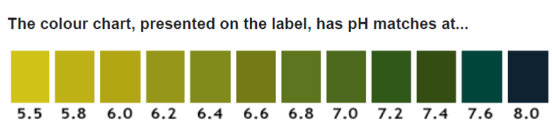


Fig. 6. Colour chart of the Hydrion's pH plastic test strips.

In addition, a small test set was created by dipping physical test strips into solutions with different pH values of 8.02, 7.57, 6.86, 6.00, and 5.76, shown in Table VI. The pH values of each solution were measured using an electronic pH probe, and images of each test strip were captured. Each image was analyzed for RGB profiling, which provides information about the intensity of each color channel (Red, Green, and Blue) in the image, and the obtained RGB profile was utilized as a data point for the respective pH. For this pilot study, we employed the default KNN model which utilizes all the default parameters offered by the *scikit-learn* library, and train it on our artificial dataset. KNN was chosen due to our understanding of its classification mechanics. The trained model was then used to predict the pH values based on the RGB profiles of individual physical test strip images. The outcomes of the predicted pH values are presented in Table VI.

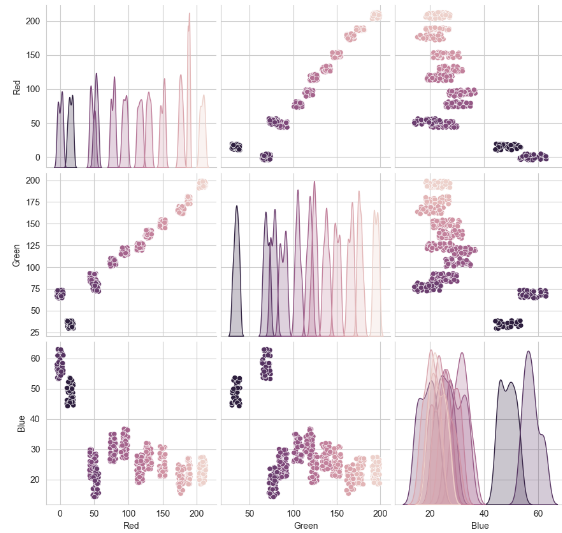


Fig. 7. Visualization of the Artificial Dataset. The axis of each subplot corresponds to RGB values for a specific pH value range, ranging from 5.5 to 8.0, as derived from the colour chart in Fig. 6.

In Table VI, the pH values predicted by the KNN model are comparable to the actual solution pH values. This indicates the promising use of machine learning models for pH classification based on test strip color recognition. However, further research is needed to investigate the feasibility of using machine learning for this purpose, which requires a robust training dataset from physical pH test strip images. If successful, the classification can potentially be expanded to determine BUN levels.

TABLE VI. Results of pH Prediction Using a Default KNN Model from *scikit-learn* Library Trained on Artificial Dataset

Actual pH Value	Predicted pH Value
8.02	8.0
7.57	8.0
6.86	7.4
6.00	6.6
5.76	6.2

V. Conclusion and Future Work

The application of machine learning for pH classification based on the color of pH test strips shows promise. By using a selected dataset from Kaggle and testing on several supervised machine learning models, the accuracy for correctly identifying pH ranges from approximately 83.33% to 86.36%. However, accuracy varies between test cases due to the random partitioning of the dataset and the presence of outliers. The performance of the machine learning models is affected by the presence of outliers or the lack of hyperplanes between pH RGB profiles. In the pilot experiment using a KNN model trained on the proposed artificial dataset with more robust boundaries, which resulted in a significant improvement in accuracy. This finding provides evidence that defining more robust boundaries or distinct hyperplanes between RGB profiles for specific pH values can enhance the performance of machine learning models.

Due to the limited availability of datasets for this study, including the Kaggle dataset that contains outliers and lacks clear boundaries between pH RGB profiles, our future studies can focus on creating and evaluating larger and more robust datasets using physical pH test strip color images. By reducing the effects of outliers and minimizing the overlapping of RGB profiles, the performance of machine learning models could be improved. Additionally, weighting the impact of individual colors (Red, Green, and Blue) on the model's performance could be examined. An ideal future direction would be to develop a mobile application that can recognize the pH value from a physical pH test strip provided by patients.

REFERENCES

[1] Centers for Disease Control and Prevention. Chronic Kidney Disease in the United States, 2021. Atlanta, GA: US Department of Health and Human Services, Centers for Disease Control and Prevention; 2021.

[2] Mayo Clinic. "Acute Kidney Failure: Symptoms and Causes", July 2020.

[3] Pooja K S, R N Shreya, Yashika BC, Sree Lakshmi M, and Mrs. Rekha B N, "Color Recognition using K-Nearest Neighbors Machine Learning classification algorithm trained with Color Histogram Features," International Research Journal of Engineering and Technology (IRJET), vol. 08(01), pp. 1935-1937, January 2021.

[4] C.S. Pundir, Seema Jakhar, and Vinay Narwal, "Determination of urea with special emphasis on biosensors: A review," Biosensors and Bioelectronics, vol. 123, pp. 36-50, 2019.

[5] Soni, Anuradha, Rajat Kumar Surana, and Sandeep Kumar Jha, "Smartphone based optical biosensor for the detection of urea in saliva," Sensors and Actuators B: Chemical, vol. 269, pp. 346-353, September 2018.

[6] Wang, Kuan-Hsun, et al., "A low-cost, portable and easy-operated salivary urea sensor for point-of-care application," Biosensors & bioelectronics, vol. 132, pp. 352-359, May 2019.

[7] Kumar, N., Kaur, N., and Gupta, D., "Red Green Blue Depth Image Classification Using Pre-Trained Deep Convolutional Neural Network," Pattern Recognition and Image Analysis, vol. 30, pp. 382-390, July 2020.

[8] Justiawan, et al., "Comparative Analysis of Color Matching System for Teeth Recognition Using Color Moment," Med Devices (Auckl), vol. 12, pp. 497-504, December 2019.

[9] Chavan, Ajinkya, "Building RGB Color Classifier: Part 1," Analytics Vidhya, September, 2020.

[10] Kaggle Dataset. "pH-recognition", 2018. https://www.kaggle.com/datasets/robjan/ph-recognition