# Classification of Fresh and Rotten Fruits in an iOS-Based Application for the Visually Impaired

Anugerah Wibisana[1] and Wahyu Nanda Nicola[2]

{wibisana@polibatam.ac.id[1], wahyunandanicola@gmail.com[2]}

Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia[1,2]

**Abstract.** Visually impaired people face various challenges, including difficulties in preparing food, particularly in selecting fresh fruits, which are essential for nutritional needs. An iOS-based application using machine learning can assist in distinguishing between fresh and rotten fruits. This research aims to develop an iOS application to classify fresh and rotten fruits for the visually impaired using Create ML for model development. The classification model, built with Create ML, is integrated into the iOS application. The model and aplication's performance are evaluated using accuracy, precision, recall, and F1 score metrics. The model achieved 99% accuracy in training, 96% in validation, and 96% in testing. When tested with 100 samples from the fruit dataset, the application achieved an overall accuracy of 93%. Testing with images outside the dataset resulted in 88% accuracy. This application aims to help visually impaired individuals identify fresh and rotten fruits, enhancing their independence.

**Keywords:** Visually Impaired, iOS Application, Create ML

## 1 Introduction

The World Health Organization (WHO) reports that at least 2.2 billion people in the world are visually impaired [1]. Visual impairments result in a variety of challenges, including difficulty in performing daily activities independently [2]. Research shows that individuals with visual impairments tend to be more prone to malnutrition and face difficulties in shopping, preparing and consuming food [3]. Vegetables and fruits themselves play a crucial role in fulfilling the nutritional needs of the human body [4]. In Indonesia, the average fruit consumption only reaches 81.14 grams per capita per day, which is still far below the WHO recommendation of 150 grams per day [5]. Visually impaired people on the other hand, face additional difficulties in identifying objects including fresh fruits, which may affect their ability to achieve adequate nutritional intake. In their daily lives, they rely heavily on the help of others to carry out their activities [6]. Therefore, technology such as smartphones can play a role in improving the independence of the visually impaired [7]. In particular, technology that can help their daily lives, including in selecting fresh fruits.

Machine learning technology has shown great potential in helping visually impaired people lead more independent lives. One of the technologies that can be used is Create ML, Apple's machine learning framework that enables the development of efficient classification models and can be integrated into iOS applications [8]. Previous research has used Create ML to build a banknote denomination recognition model for the visually impaired [9]. This shows the potential of Create ML to be used in various applications that support the needs of the visually impaired.

In terms of classifying fresh and rotten fruits, there have been several studies using various types of models and implemented in Android-based applications. For example, research conducted by Prinzky and C. Lubis on classification of fresh and rotten fruits using android- based CNN [10]. Another study by D. Rahman Sya'ban, A. Hamzah, and E. Susanti on classification of fresh and rotten fruits using CNN algorithm with Tflite as a media application of machine learning models [11]. However, in this research, the classification model will be implemented on an iOS-based application.

The application to be developed in this research aims to help visually impaired people recognize fresh and rotten fruit using the classification model from Create ML which is implemented on an iOS-based application. The selection of fruit quality is important because fresh fruit has a high nutritional content and is free from harmful microorganisms that are often found in rotten fruit [12]. The application will use the VoiceOver feature, a built-in iOS accessibility technology, which provides voice descriptions of fruit classification results so that visually impaired people can use it without direct vision. This research is expected to facilitate the daily life of the visually impaired and also contribute to the field of accessibility and health technology.

## 2 Method

This research uses a machine learning-based approach to classify fresh and rotten fruits. In order to achieve this goal, through several main stages which include data collection, data preparation, model building, model evaluation, application development and application performance evaluation. The research flow chart can be seen in Figure 1.

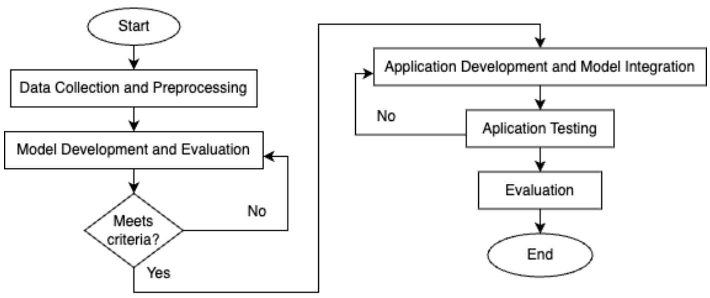



**Fig. 1.** Research Flow Chart

### 2.1 Data collection and preprocessing

This research uses data in the form of fruit images consisting of fresh apples, rotten apples, fresh oranges, rotten oranges, fresh lemons, rotten lemons, fresh bananas, and rotten bananas. Image capture is done independently using a smartphone camera with a resolution of 12 MP and an image size of 3024 x 4032. The dataset collected amounted to 5000 images with each class consisting of 500 images and divided into training data and test data with a ratio of 80:20 using the Python programming language and the Google Collaboratory platform which resulted in 4000 for train data and 1000 for test data. Image capture was carried out with burst mode with varying angles. Fruit objects were also varied such as one fruit, two fruits and three fruits. Examples of images from each class can be seen in Figure 2.

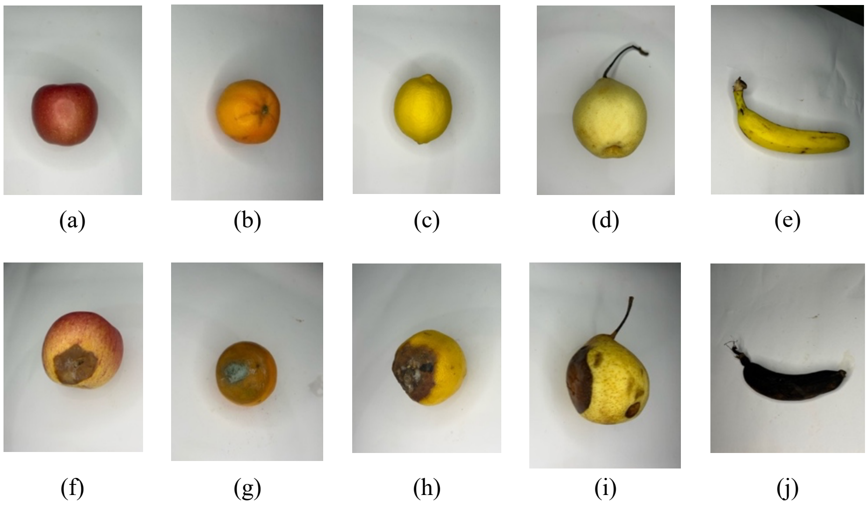



**Fig. 2.** Example images of dataset (a) Fresh Apple, (b) Fresh Orange, (c) Fresh Lemon, (d) Fresh Pear, (e) Fresh Banana, (f) Rotten Apple, (g) Rotten Orange, (h) Rotten Lemon, (i) Rotten Pear, (j) Rotten Banana.

### 2.2 Model development

In this model building stage, researchers used Create ML provided by Apple. Create ML was chosen because the results of model building from Create ML can be directly downloaded in .mlmodel format, which can be implemented into iOS-based application development. Create ML has an easy-to-use interface, and the modeling process does not require programming. In the process of making this model, researchers used a 13- inch Macbook Pro, M2 2022. In Create ML, there are several templates, researchers chose the image classification template and the next step is to create a new project and input the previously prepared dataset, including training data and test data, while for validation data using the auto split feature. The researcher then set the parameters to be used such as feature extractor, iteration and data augmentation. The Create ML display can be seen in Figure 3.

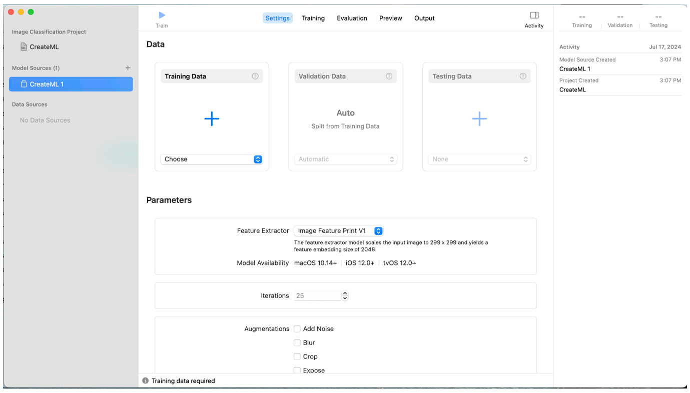


**Fig. 3.** Create ML

## 2.3 Model evaluation

The model evaluation stage uses four main metrics to assess model performance: accuracy, precision, recall, and F1 score. Accuracy is used to measure the extent to which the model's predictions are close to the true value, i.e. it represents the percentage of correct classifications [13]. Precision is the match between the user's data query and the results provided by the system [14]. Recall is defined as the ratio between the number of relevant items selected to the total relevant items available [15]. F1-Score is the harmonic mean between precision value and recall value [16]. Accuracy measurement can be seen in equation (1), precision measurement formula using equation (2), recall measurement using equation (3) and F1 Score measurement using equation (4).

$$\text{accuracy} = \frac{\sum \text{TP}}{\text{n}} \tag{1}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

$$\text{F1} = 2 * \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \tag{4}$$

Variable Description:

- TP (True Positive) = Number of correctly classified samples.
- n = The total amount of predicted data.
- FN (False Negative) = The number of samples that are misclassified as another class.
- FP (False Positive) = The number of samples incorrectly classified as this class.

## 2.4 Application development

Application development consists of design and program development. Designing is done in the Figma application and program creation is done in the Xcode application. Based on best practices on the Apple Developer Documentation web, simplicity is something that must be prioritized by creating a familiar design and consistent interactions that make complex tasks simple and easy to do [17]. In programming, several main frameworks are required, namely UIKit, Vision, and Core ML, which use the Swift programming language. UIKit is used to build and manage the application's user interface, allowing user interaction with features such as image capture using the camera and display of prediction results. Vision, as a framework provided by Apple, works for image analysis, facilitating tasks such as object detection and classification. Core ML enables the integration of machine learning models into applications. In this app Core ML uses APIs to make predictions based on user data directly on the device. It optimizes performance by leveraging CPU, GPU, and Neural Engine, while reducing memory and power consumption. By running the models on the device, Core ML keeps user data private and application responsiveness high [18]. For the flow of using the application, it starts from the user opening the application, then opens the camera in the application, takes a picture of the object, uses the image, displays the results and the user hears the detection results read by VoiceOver. For more details can be seen in Figure 4.

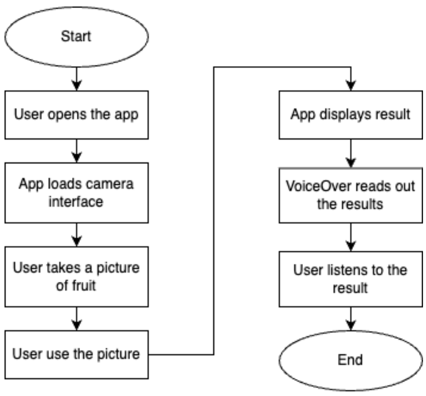



**Fig. 4.** Flowchart of Application Usage

### 2.5 Application testing and evaluation

After the application program has been built, the application can be tested by running it on an iPhone device as illustrated in Figure 5. This application test is carried out to evaluate the accuracy level of the application that has been integrated with the model. From this test, the accuracy will be calculated using equation (1), precision measurement using equation (2), recall measurement using equation (3) and F1 Score measurement using equation (4) which are listed in 2.3.

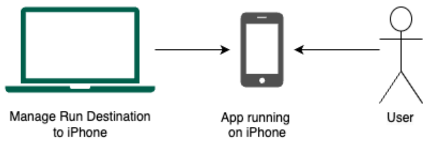



**Fig. 5.** Running the app on an iPhone for testing

## 3 Results and Discussion

### 3.1 Parameter testing on Create ML

In Create ML there are parameters that can be set such as Feature Extractor, iteration and augmentation. For Feature Extractor, researchers used Image Feature Print V1, because the model can be used on iOS 12.0+, macOS 10.14+, tvOS 12.0+. For the selection of iteration parameters, 3 experiments were carried out on different models. The following comparison is in table 1 and then for comparison of the use of augmentation can be seen in table 2.

**Table 1.** Comparison of trial iteration values

| Number of iterations | Training accuracy |
|---|---|
| 10 | 90% |
| 25 | 99% |
| 50 | 98% |

**Table 2.** Comparison of the use of augmentations in Create ML

| Augmentations | | | | | | Number of iterations | Training accuracy | Testing accuracy |
|---|---|---|---|---|---|---|---|---|
| Add Noise | Blur | Crop | Expose | Flip | Rotate | | | |
| Yes | Yes | Yes | Yes | Yes | Yes | 25 | 99% | 96% |
| No | No | No | No | No | No | 25 | 100% | 92% |

In table 1, it can be seen that by starting training with iterations of 25, higher accuracy is obtained, so researchers use iterations of 25. Based on table 2, the comparison of training accuracy values is only 1% difference, but in testing accuracy by using augmentation the accuracy value is 4% higher, so researchers decided to use all augmentation.

### 3.2 Model development results on Create ML

After parameter testing, researchers decided to set the iterations at 25, use all augmentations and use Image Feature Print V1. The modeling results can be seen in table 3 and the accuracy graph can be seen in Figure 6.

**Table 3**. Accuracy results on Create ML

| Process in Create ML | Amount of data | Data per class | Accuracy |
|---|---|---|---|
| Training | 7600 | 760 | 99% |
| Validation | 400 | 40 | 96% |
| Testing | 1000 | 100 | 96% |

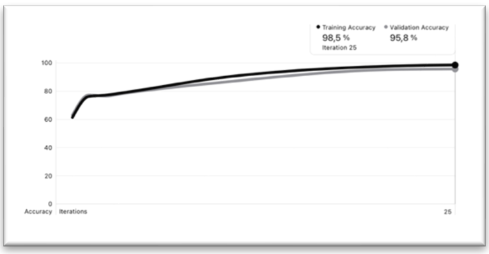



**Fig. 6**. Training and validation accuracy chart

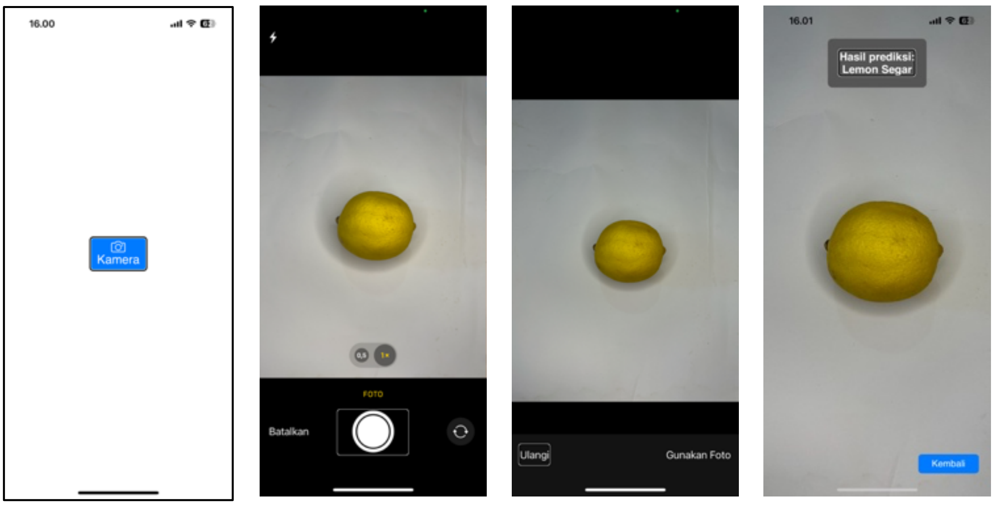
In Table 3, the resulting training accuracy is 99% with a total of 7600 data which is generated from augmentation. Validation accuracy of 96% with a total of 400 data generated from autosplit features. Testing accuracy of 96%, the data used amounted to 1000 images derived from the division results in the previous stage. Figure 6 is a graph displayed from Create ML, on the graph the training accuracy is 98.5% and the validation accuracy is 95.8%. This value is the value before rounding which is 99% for training and 96% for validation. Create ML also displays the precision, recall and F1 score values for each class in the training process as can be seen in table 4.

**Table 4**. Model Training Metrics

| No. | Class | Number of samples | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 1. | Apel Busuk | 760 | 97% | 97% | 0,97 |
| 2. | Apel Segar | 760 | 98% | 98% | 0,98 |
| 3. | Jeruk Busuk | 760 | 99% | 96% | 0,97 |
| 4. | Jeruk Segar | 760 | 97% | 97% | 0,97 |
| 5. | Lemon Busuk | 760 | 99% | 98% | 0,99 |
| 6. | Lemon Segar | 760 | 97% | 99% | 0,98 |
| 7. | Pir Busuk | 760 | 99% | 99% | 0,99 |
| 8. | Pir Segar | 760 | 100% | 100% | 1,0 |
| 9. | Pisang Busuk | 760 | 100% | 100% | 1,0 |
| 10. | Pisang Segar | 760 | 100% | 100% | 1,0 |

### 3.3 Application development

The application is built using the swift programming language. The model from CreateML can be downloaded in .mlmodel format which is a model format for iOS-based application development. Then the model is integrated into the application being developed. To see the performance of the application in the form of accuracy in the application, the testing process is carried out through a smartphone directly. Developing the user interface using the UIkit framework and to make sure the app can be read by the VoiceOver feature, it is important to set the accessibility section to enabled in the Identity inspector and mark some traits boxes. The results of the user interface can be seen in figure 7.




(a) (b) (c) (d)

**Fig 7.** User Interface (a) Screen for opening the camera, (b) Camera screen, (c) Screen for using photos, (d) Prediction result screen

**Application usage with VoiceOver.** There are several gestures that can be used, such as double-tap to activate the selected item, three-finger swipe to scroll and so on. When VoiceOver is turned on, a VoiceOver cursor will also appear. This box-shaped VoiceOver cursor is used to highlight the element that VoiceOver is focusing on, helping users to know which element is being selected or read by VoiceOver. In Figure 7 (a) the user double-taps with one finger on the “Kamera” button to open the camera, in Figure 7 (b) double- taps the "Ambil Gambar" button with one finger to take a picture, in Figure 7 (c) the user taps twice the “Ulangi” button to repeat take a picture, if user want to use an image, swipe from left to right with one finger and then double- taps the "Gunakan Foto" and in Figure 7 (d) VoiceOver cursor will read out the prediction result when cursor on the prediction result, in the picture above it will read "Hasil prediksi: Lemon Segar" means the result is fresh lemon.

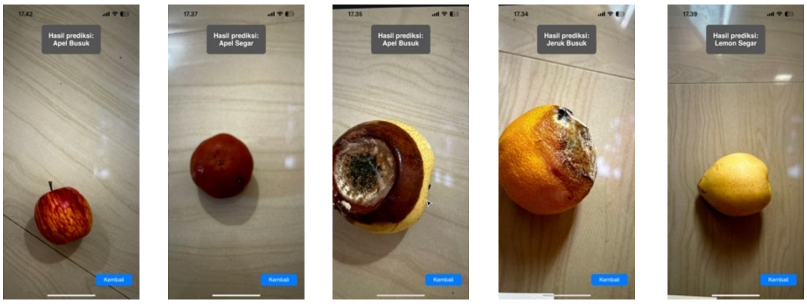
### 3.4 Application Testing

**Application testing on sample fruit datasets.** This testing stage is carried out on 100 samples from 10 classes, each consisting of 10 samples. The test results for each class are summarized in Table 5.

**Table 5**. Aplication Testing Metrics

| No. | Class | Number of samples | TP | FP | FN | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|---|---|
| 1. | Rotten Apple | 10 | 9 | 3 | 1 | 0.75 | 0.90 | 0.818 |
| 2. | Fresh Apple | 10 | 9 | 2 | 1 | 0.82 | 0.90 | 0.857 |
| 3. | Rotten Orange | 10 | 9 | 1 | 1 | 0.90 | 0.90 | 0.90 |
| 4. | Fresh Orange | 10 | 10 | 0 | 0 | 1.00 | 1.00 | 1.00 |
| 5. | Rotten Lemon | 10 | 8 | 0 | 2 | 1.00 | 0.80 | 0.889 |
| 6. | Fresh Lemon | 10 | 10 | 1 | 0 | 0.91 | 1.00 | 0.952 |
| 7. | Rotten Pear | 10 | 9 | 0 | 1 | 1.00 | 0.90 | 0.947 |
| 8. | Fresh Pear | 10 | 9 | 0 | 1 | 1.00 | 0.90 | 0.947 |
| 9. | Rotten Banana | 10 | 10 | 0 | 0 | 1.00 | 1.00 | 1.00 |
| 10. | Fresh Banana | 10 | 10 | 0 | 0 | 1.00 | 1.00 | 1.00 |

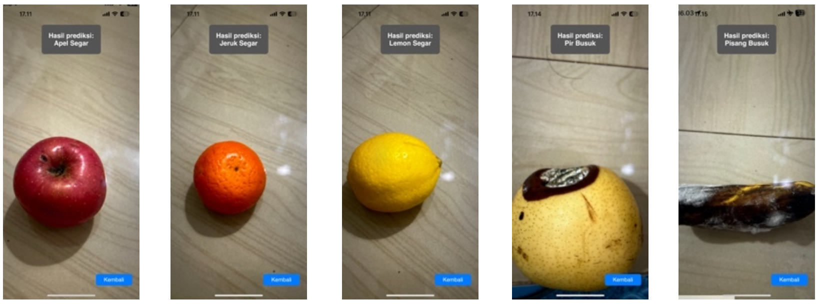
Based on Table 5, the overall accuracy of testing the application on the fruit used as a dataset is 93%, calculated as the total number of correct predictions ($\sum TP$=93) divided by the total number of samples tested (n=100). The application is able to classify the Rotten Banana and Fresh Banana classes with precision, recall, and F1 score values of 1.00, reflecting the absence of errors in the classification of samples for both classes. Lower performance was seen in the Rotten Apple class with a precision of 0.75 due to high false positives (FP), indicating that many items from other classes were misclassified as Rotten Apples. In addition, Rotten Lemon showed the lowest recall of 0.80 due to the presence of false negatives (FN), meaning some Rotten Lemon samples were misclassified to other classes. To see some examples of misclassification, can be seen in Figure 8.




(a) (b) (c) (d) (e)

**Fig. 8.** Some examples of incorrect classification results (a) Fresh Apple as Rotten Apple, (b) Rotten Orange as Fresh Apple, (c) Rotten Pear as Rotten Apple, (d) Rotten Lemon as Rotten Orange, (e) Fresh Pear as Fresh Lemon.

Based on Figure 8, it can be seen that Figure 8 (a) is a Fresh Apple but is predicted as a Rotten Apple due to the uneven color of the apple surface. Figure 8 (b) Rotten Orange is predicted as a Fresh Apple due to the similarity in shape and color. Figure 8 (c) Rotten Pear is predicted as a Rotten Apple, due to the similarity of the image with a Rotten Apple. Figure 8 (d) Rotten Lemon is predicted as Rotten Orange due to the color of the lemon turning orange caused by the decay process. And finally, image 8 (e) Fresh Pear is predicted as Fresh Lemon due to the similarity in color and the absence of pear stalks and the angle of shooting also affects the similarity with Fresh Lemon. In Figure 9 here are some examples of correct prediction results.




(a) (b) (c) (d) (e)

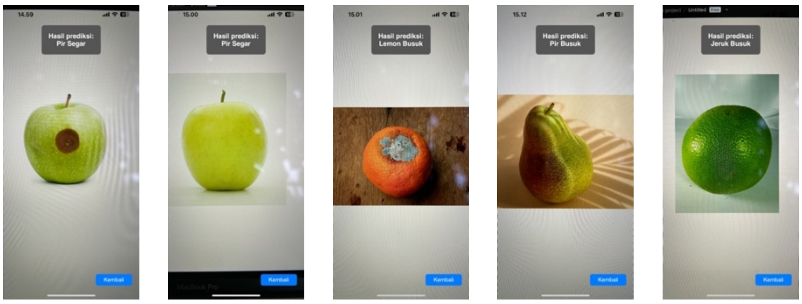
**Fig. 9.** Some examples of correct classification results (a) Fresh Apple, (b) Fresh Orange, (c) Fresh Lemon, (d) Rotten Pear, (e) Rotten Banana

**Application testing on fruit samples outside the dataset.** This test is carried out on images of fruits found on the internet which aims to measure the extent to which the application can classify images outside of the dataset. There are 100 test samples, the following test results are in table 6.

**Table 6**. Aplication testing metrics

| No. | Class | Number of samples | TP | FP | FN | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|---|---|
| 1. | Rotten Apple | 10 | 8 | 2 | 2 | 0.80 | 0.80 | 0.80 |
| 2. | Fresh Apple | 10 | 9 | 0 | 1 | 1.00 | 0.90 | 0.947 |
| 3. | Rotten Orange | 10 | 8 | 2 | 2 | 0.80 | 0.80 | 0.80 |
| 4. | Fresh Orange | 10 | 8 | 2 | 2 | 0.80 | 0.80 | 0.80 |
| 5. | Rotten Lemon | 10 | 9 | 2 | 1 | 0.82 | 0.90 | 0.857 |
| 6. | Fresh Lemon | 10 | 9 | 0 | 1 | 1.00 | 0.90 | 0.947 |
| 7. | Rotten Pear | 10 | 9 | 1 | 1 | 0.90 | 0.90 | 0.90 |
| 8. | Fresh Pear | 10 | 9 | 3 | 1 | 0.75 | 0.90 | 0.818 |
| 9. | Rotten Banana | 10 | 10 | 0 | 0 | 1.00 | 1.00 | 1.00 |
| 10. | Fresh Banana | 10 | 9 | 0 | 1 | 1.00 | 0.90 | 0.947 |

Based on Table 6, the overall accuracy of testing the application on fruits outside of this dataset was 88%, calculated as the total number of correct predictions ($\sum TP$=88) divided by the total number of samples tested (n=100). The application was able to classify the Rotten Banana class with precision, recall, and F1 score values of 1.00, reflecting the absence of errors in the classification of samples for that class. Lower performance was seen in the Rotten Apple and Rotten Orange classes with a precision and recall of 0.80 due to high false positives (FP) and false negatives (FN), indicating that many items from other classes were misclassified as Rotten Apple and Rotten Orange. In addition, the Fresh Orange class also had a low precision and recall of 0.80, indicating the model had difficulty in correctly classifying the Fresh Orange samples. Rotten Lemon showed a recall of 0.90 but a slightly lower precision of 0.82 due to the presence of false negatives, meaning some Rotten Lemon samples were misclassified to other classes. The Fresh Pear class has a precision of 0.75 despite a recall of 0.90, indicating that some Fresh Pear predictions were misclassified to other classes. To see some examples of misclassification, can be seen in Figure 10.




(a) (b) (c) (d) (e)

**Fig. 10.** Some examples of incorrect classification results (a) Rotten Apple as Fresh Pear, (b) Fresh Apple as Fresh Pear, (c) Rotten Orange as Rotten Lemon, (d) Fresh Pear as Rotten Pear, (e) Fresh Orange as Rotten Orange.

Based on Figure 10 (a) Fresh Apples are predicted as Fresh Pears because the fruit is not really rotten and has some similarity in shape with Fresh Pears. Figure 10 (b) Fresh Apples are predicted as Fresh Pears due to similarity in shape and green apples are not found in the dataset.

Figure 10 (c) Rotten Orange as Fresh Orange due to the less visible rotten part of the fruit. Figure 10 (d) Fresh Pear is predicted as a Rotten Pear because pears with green and reddish colors are not found in the dataset plus it can be seen in the image that the position of the fruit is exposed to shadows which is a factor in the wrong prediction. Figure 10 (e) Fresh green oranges are classified as Rotten Oranges because Green Oranges are not present in the dataset.

## 4 Conclusion

This research develops an iOS-based application for the classification of fresh and rotten fruits aimed at helping visually impaired people. This application uses a machine learning model built with Create ML and can be used with the VoiceOver feature. The model development results achieved 99% accuracy in training, 96% in validation, and 96% in testing. In addition, testing through the application with 100 testing on fruits used as datasets resulted in an overall accuracy of 93% and testing the application on images outside the dataset obtained an accuracy of 88%, There is a 5% decrease due to testing samples that are images outside of the dataset and testing fruit images that vary such as testing on Fresh Green Oranges which are classified as Rotten Oranges because green oranges are not found in the dataset, so to deal with this in the future, more varied fruit images can be added to the dataset, besides that the angle of taking pictures and the position of placing the fruit also affects the prediction results. Overall, the developed application shows the potential to increase the independence of the visually impaired in choosing fresh fruits. By using the VoiceOver feature, the application provides a voice description of the classification results, so that visually impaired people can operate the application without the need for direct vision. This innovation is expected to help visually impaired people in selecting fresh and rotten fruits and improve their quality of life.

## Acknowledgements.

The author would like to thank the supervisor, Mr. Anugerah Wibisana, S.Tr.T., M.Tr.T, for his guidance and direction during the research. The author would also like to thank family and friends who have been willing to take the time to support in this research.

## References

[1] "Blindness and vision impairment." Accessed: Jul. 12, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] C. TSHUMA, N. NTOMBELA, and H. C. VAN WIJK, "Challenges and Coping Strategies of the Visually Impaired Adults: a Brief Exploratory Systematic Literature Review," *Prizren Soc. Sci. J.*, vol. 6, no. 2, pp. 71–80, 2022, doi: 10.32936/pssj.v6i2.301.

[3] N. Jones, H. E. Bartlett, and R. Cooke, "An analysis of the impact of visual impairment on activities of daily living and vision-related quality of life in a visually impaired adult population," *Br. J. Vis. Impair.*, vol. 37, no. 1, pp. 50–63, 2019, doi: 10.1177/0264619618814071.

[4] W. dan T. F. Hariani, "Edukasi Pentingnya Konsumsi Sayur Dan Buah Pada Siswa Sekolah Dasar Negeri 2 Soropia Kecamatan Soropia Kabupaten Konawe," *J. Pengabdi. Masy.*, vol. 01, no. 02, pp. 36–47, 2021.

[5] H. Limanseto, "Terus Dorong Peningkatan Konsumsi Buah Nusantara, Pemerintah Gelar

Kembali Gelar Buah Nusantara (GBN) ke-7 Tahun 2022 - Kementerian Koordinator Bidang Perekonomian Republik Indonesia," Kementrian Koordinator Bidang Perekonomian Republik Indonesia. [Online]. Available: https://www.ekon.go.id/publikasi/detail/4450/terus-dorong-peningkatan-konsumsi-buah-nusantara-pemerintah-gelar-kembali-gelar-buah-nusantara-gbn-ke-7-tahun-2022

[6] J. Y. Mambu, E. Anderson, A. Wahyudi, G. Keyeh, and B. Dajoh, "Blind Reader: An Object Identification Mobile- based Application for the Blind using Augmented Reality Detection," *2019 1st Int. Conf. Cybern. Intell. Syst. ICORIS 2019*, vol. 1, no. August, pp. 138–141, 2019, doi: 10.1109/ICORIS.2019.8874906.

[7] L. Hidayat, "Assistive Technology Pada Aplikasi," *J. Exponential (Education Except. Child.*, vol. 1, no. 2, pp. 144–152, 2020.

[8] "Create ML | Apple Developer Documentation." Accessed: Jun. 19, 2024. [Online]. Available: https://developer.apple.com/documentation/createml

[9] C. Suardi, D. Sundoro, K. Syariati, and R. L. Lordianto, "Bank Note Value Checker Based on Image Recognition," *J. Technopreneur*, vol. 11, no. 1, pp. 24–29, 2023, doi: 10.30869/jtech.v11i1.1161.

[10] Prinzky and C. Lubis, "Klasifikasi Buah Segar Dan Busuk Menggunakan Convolutional Neural Network Berbasis Android," *J. Ilmu Komput. dan Sist. Inf.*, vol. 10, no. 2, pp. 1–5, 2022, doi: 10.24912/jiksi.v10i2.22551.

[11] D. Rahman Sya'ban, A. Hamzah, and E. Susanti, "Klasifikasi Buah Segar Dan Busuk Menggunakan Algoritma Convolutional Neural Network Dengan Tflite Sebagai Media Penerapan Model Machine Learning," *Pros. Snast*, no. November, pp. F7-16, 2022, doi: 10.34151/prosidingsnast.v8i1.4180.

[12] R. Handoyo, D. Erny Herwindiati, T. Sutrisno, U. Tarumanagara, and K. Kunci, "Klasifikasi Buah Busuk Dan Buah Matang Berdasarkan Data Image Menggunakan Mahalanobis Distance," *Humantech J. Ilm. Multi Disiplin Indones.*, vol. 2, no. 4, pp. 640–654, 2023.

[13] M. A. Rasyidi and T. Bariyah, "Batik pattern recognition using convolutional neural network," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1430–1437, 2020, doi: 10.11591/eei.v9i4.2385.

[14] V. Oktaviana Yamin *et al.*, "Penerapan Naïve Bayes Classifier dengan Algoritma Nazief dan Adriani Untuk Deteksi Hoaks," vol. 1, no. 1, pp. 1–5, 2023.

[15] D. Pradana, M. Luthfi Alghifari, M. Farhan Juna, and D. Palaguna, "Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network," *Indones. J. Data Sci.*, vol. 3, no. 2, pp. 55–60, 2022, doi: 10.56705/ijodas.v3i2.35.

[16] A. M. Argina, "Penerapan Metode Klasifikasi K-Nearest Neigbor pada Dataset Penderita Penyakit Diabetes," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 29–33, 2020, doi: 10.33096/ijodas.v1i2.11.

[17] "Accessibility | Apple Developer Documentation." Accessed: Jun. 25, 2024. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/accessibility

[18] "Core ML | Apple Developer Documentation." Accessed: Jul. 10, 2024. [Online]. Available: https://developer.apple.com/documentation/coreml