Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.3, June 2025, pp. 863~867
e-ISSN: 2548-6861

Development of ViScan: A Mobile Application for Skin Cancer Detection Using Ionic Framework and YOLOv10x

Alif Agsakli Haresta [1]*, Cinantya Paramita [2], William Dwiputra Tjahtjono [3]***
* Teknik Informatika, Universitas Dian Nuswantoro
alifagsaklih4@gmail.com [1], cinantya.paramita@dinus.ac.id [2], william.dwiputra37@gmail.com [3]

Article Info

Article history:

Received 2025-04-16
Revised 2025-05-26
Accepted 2025-06-04

Keyword:

Skin Cancer Detection,
Mobile Application,
YOLOv10x,
Ionic Framework,
Mobile Health Application.

ABSTRACT

Skin cancer is a common global health issue, with the number of cases continuing to rise worldwide. Early detection is crucial for improving patient outcomes, but traditional detection methods often require significant time, cost, and medical expertise. To address this challenge, this research focuses on developing a mobile application that leverages deep learning, specifically the YOLOv10x model, to enable fast and accurate detection of skin lesions. This application aims to provide an easy-to-use platform for self-monitoring skin health, particularly for individuals in remote areas with limited access to medical facilities. The system uses the HAM10000 dataset, which consists of a diverse collection of dermoscopy images of skin lesions, to train the YOLOv10x object detection model for real-time detection on mobile devices. By leveraging TensorFlow.js and Node.js, the model processes skin images and provides real-time results with precision and efficiency. The mobile application, developed using the Ionic Framework, ensures cross-platform compatibility and a responsive, intuitive user interface. System performance was evaluated using key metrics such as Precision (84.2%), Recall (86.3%), mAP (89.2%), and F1 Score (85.2%), demonstrating its effectiveness in early skin cancer detection. The potential of this application extends beyond detection, contributing to society by raising awareness and offering an accessible, low-cost screening solution.

I. PENDAHULUAN

Skin cancer is one of the most common types of cancer in the world, with the number of cases increasing over time [1]. Early detection of skin cancer is crucial to increase the patient's chances of recovery, but conventional detection methods often require a long time, high costs, and in-depth medical expertise [2]. Although a direct dermatological examination can detect suspicious skin lesions, limited access to medical personnel and expensive medical technology means that many people do not get routine examinations [3]. Therefore, it is important to develop technology that enables independent and faster detection of skin cancer.

Along with technological advances, especially in the fields of machine learning and deep learning, a number of studies have shown the extraordinary potential of this technology in detection various diseases, including skin cancer. One popular technique is the use of deep learning to analyze medical images, where the You Only Look Once (YOLO) model has proven effective in detecting objects in images in real-time with high accuracy. YOLO, which was originally developed for the detection of objects in images in general, has now been applied in various medical applications, including for detection skin lesions. One of the datasets widely used in dermatology research is HAM10000, which consists of diverse dermatoscopic images and can be used to train machine learning models in detecting skin cancer [4].

Although many studies have examined the application of YOLO in the medical field, the application of this model in a mobile application that is widely accessible to the general public is still very limited. Mobile applications have great potential to improve the accessibility and ease of skin lesion detection, especially for individuals living in areas with limited medical facilities [5], [6]. This kind of application allows users to monitor their skin health regularly and quickly, without the need to rely on expensive and time-consuming medical visits.

This research aims to develop a mobile application that utilizes machine learning technology, specifically the YOLO model, to detect skin lesions automatically and accurately. This application is designed to identify suspicious skin lesions using the HAM10000 dataset, which contains dermatoscopic images of various types of skin lesions. By utilizing YOLO's real-time object detection performance, this application is expected to provide fast and accurate detection results, which can be directly accessed by users through their mobile devices [7]. The main objective of this research is to develop a mobile application that can detect skin lesions automatically and accurately, using machine learning technology and object detection models such as YOLO [8]. This research aims to evaluate the effectiveness of mobile applications in providing fast and reliable detection results for users, as well as optimizing the performance of the application so that it can run well on mobile devices with limited resources.

The benefits of this research are extensive, both technically, socially, and in terms of health. Technically, this research will contribute to the development of artificial intelligence-based health applications that can be accessed by the public. Socially, this application can facilitate the community in conducting early detection of skin lesions without the need to rely on professional medical examinations, which will be very helpful, especially in areas with limited access to medical facilities. In terms of health, this application has the potential to raise public awareness about the importance of routine skin health checks, as well as providing a cheaper and more accessible solution for the early detection of skin cancer.

II. METHOD

A. *Development Tools*

The development of this mobile application involves various tools that support the process of developing, testing, and maintaining applications efficiently. Visual Studio Code (VSCode) is used as the main text editor for writing and editing application source code, supporting programming languages such as JavaScript, HTML, CSS, and TypeScript which are at the core of Ionic-based application development [9], [10]. VSCode also offers plugins and extensions to improve application productivity and debugging. For testing applications on Android devices, Android Studio is used, which provides emulators for various devices and features to analyze application performance, including power and memory consumption, as well as optimizing performance on low- spec devices [11]. Google Chrome is used to test web-based applications and to debug applications using Chrome DevTools, which allows developers to check the console, network, memory, and application performance during development. To build a cross-platform mobile application, the Ionic Framework was chosen for its ability to create applications that can run on Android and iOS using the same source code, namely standard web technologies such as HTML, CSS, and JavaScript [12]. On the server side, Node.js acts as a runtime server that allows applications to function efficiently and responsively. With its non-blocking capabilities and real-time data processing, Node.js can handle user requests quickly, including sending images for processing. Node.js works together with REST API, which enables front-end and backend server communication [13]. Through REST API, applications can send requests, such as sending images for processing, and receive analysis results from the server.

In the database section, PostgreSQL was chosen as the database system to store medical data securely and in a structured manner. As a relational database management system, PostgreSQL ensures the reliability and integrity of medical data, such as skin cancer detection results, which are stored and can be accessed again when needed [14]. With a reliable database system, the application can maintain the quality and security of user data, which is important in the context of health applications. For the skin cancer detection process, the application relies on YOLOv10x, a deep learning model used for object detection in images [15]. YOLOv10x works by analyzing images of a patient's skin cancer, detecting abnormalities or characteristics of skin cancer quickly and accurately. The advantage of YOLOv10x is its ability to process images in a short time, which enables real-time detection, which is very useful in providing initial screening results for users [16].

TensorFlow.js, which is a JavaScript-based machine learning library, is used to run the YOLOv10x model in the application. By using TensorFlow.js, the application can run deep learning models directly on the user's device, which allows detection to be done faster without the need to rely on the server to process images [17]. This also enables real- time image processing on mobile devices, which is very important to provide fast and efficient detection results to users. On the design side, Figma is used to design the application UI/UX, create wireframes, prototypes, and visual designs of the application, and ensure a simple and intuitive interface [18]. The main dataset used in this study is HAM10000, which contains dermatoscopic images with labels of skin lesion types, which are used to train YOLO models in detecting skin lesions. Google Cloud Storage ensures scalable data storage, enabling applications to handle image and data storage without burdening user devices [19].

B. *Visualization Process*

The application development process is presented in Figure 1 workflow diagram which shows how all these components work together to detect skin cancer. This diagram will show the flow of data from image capture by the application, image processing using YOLOv10x for detection, sending data to the server via REST API, storing results in PostgreSQL, and finally displaying the detection results to the user through an interface built with Ionic.

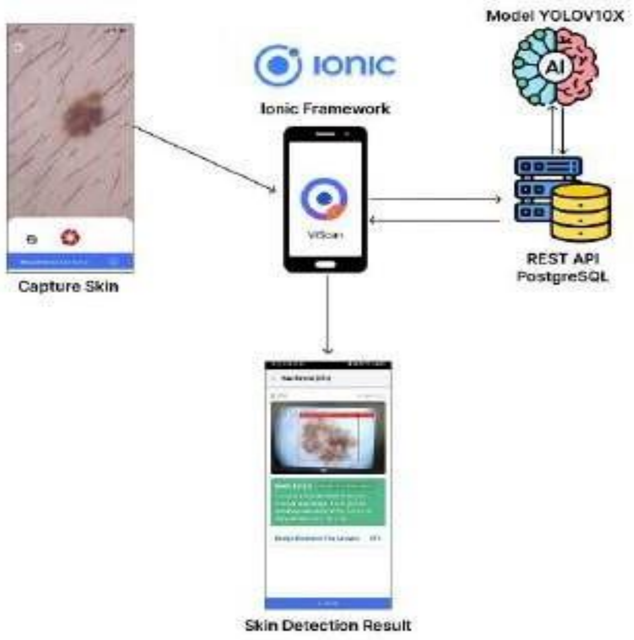

Figure 1 Skin Cancer Detection Workflow

C. Dataset and Preprocessing

The HAM10000 dataset consists of 10,015 dermoscopic images classified into seven categories of skin lesions [20]. Information regarding these seven classes is explicitly listed in the dataset configuration file and is used as the main reference in the data labeling process. Due to the uneven distribution of classes, an undersampling strategy is carried out on the class with the most data and oversampling on the class with the least data [21]. Oversampling is done through data duplication and augmentation techniques, such as rotation, flipping, zooming, and brightness level adjustment [22]. This effort resulted in an even number of images, 500 images per class, with a total of 3,500 images. The dataset was then uploaded to Roboflow for annotation and object detection- based techniques. The dataset was divided into three subsets: 80% for training, 10% for testing, and 10% for validation. In the pre-processing stage, the image is resized, adjusted to the input dimensions, and an automatic orientation feature is applied. With this class balancing and data standardization, the workflow supports accurate, unbiased model training that is in line with best practices in medical imaging research. The model's performance was evaluated using mean average precision (mAP), achieving an 89.2% mAP, along with the following metrics: Precision (84.2%), Recall (86.3%), and F1 Score (85.2%), demonstrating strong accuracy in skin lesion detection for pre-screening.

D. Deployment Process

This research uses the YOLOv10x Artificial Intelligence model, which is initially translated into TensorFlow.js format to ensure adequate functionality in the Node.js environment. After the conversion procedure, the model file is uploaded to Google Cloud Storage to be accessed and loaded by TensorFlow.js via Node.js-based REST API. After the user uploads an image using the REST API, the Node.js server, built with the Express.js framework and the Multer module, will manage the image upload procedure and start the processing workflow. The obtained image is then scaled to 224x224 pixels during the pre-processing phase to match the input requirements of the YOLOv10x model. Furthermore, the model analyzes the image to produce detection outputs, including confidence values, bounding box coordinates, and detection labels that indicate seven categories of skin lesions, namely: vascular lesions (VASC), actinic keratosis (AKIEC), basal cell carcinoma (BCC), benign keratosis-like lesions (BKL), dermatofibroma (DF), melanocytic nevi (NV), and melanoma (MEL). The model results are used by the server to insert bounding boxes and labels into the original image, producing a detection image that is then uploaded to Google Cloud Storage for secure access. Furthermore, detection data including user ID, diagnosis type, original image path, annotation image, confidence score, and bounding box coordinates (x min, y min, x max, y max) are stored systematically in the “user report” table in the PostgreSQL database. The REST API uses the HTTPS protocol and JWT (JSON Web Token) authentication to ensure data security during transmission. If problems occur during image processing or model interaction, the REST API will send an error message to the user [23]. This integration of TensorFlow.js, Node.js, and PostgreSQL forms a secure, efficient, and reliable backend system for skin cancer detection, making it easier for users to obtain detection information quickly and efficiently.

III. RESULT AND DISCUSSION

The ViScan application was developed using the Ionic Framework, a web-based platform that supports the creation of cross-platform applications for Android and iOS devices. This framework utilizes web technologies such as HTML, CSS, and JavaScript, and Ionic, which allows the user interface to be responsive, intuitive, and consistent across various devices. The application consists of several main. displays to support its functions. The registration view allows users to register by filling out a form with their full name, username, phone number, password, and confirmation password. The login view provides a form for logging in with a username and password. The home view provides navigation to the skin detection page, medical record page, and education page. The skin cancer detection camera view is used to scan skin lesions using the device's camera, and the image processing view shows a loading indicator while the image is being processed. The information view displays the detection results in the form of bounding boxes and labeled images, as well as the accuracy level and detection status of the cancer type. This user interface layout is visualized in Figure 2, which shows the relationship between the main views of the application.

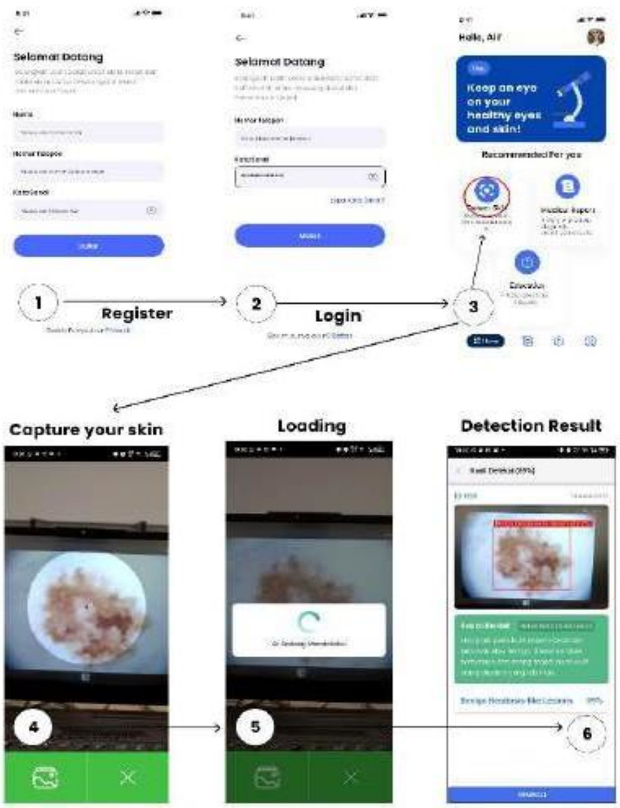

Figure 2 User Workflow

Figure 3 visually displays the skin cancer detection process, starting from the use of a camera to scan skin lesions to the analysis results in the form of images with bounding boxes and labels detection the type of cancer. This visualization illustrates how the application utilizes the YOLOv10x model to detect skin cancer with high accuracy and provide an informative experience for users. With a combination of modern technology and a structured interface, this application makes it easy for users to detect and monitor their skin condition effectively and independently.

Figure 2 Score of the user's skin cancer detection result

To ensure that the designed UI elements work according to the expected functionality, testing is carried out using the black-box testing method. This test evaluates the functionality of the application without examining the internal code structure. The main focus of this test is the navigation between pages and the clarity of the detection results display, considering that these two aspects are key to providing an optimal user experience. The test scenario covers important features of the application, such as the registration process, login, navigation to the detection page, and display of detection results. The expected test results will be compared with the actual results to ensure that each function works as planned. This test ensures that the application is not only well-used but also provides a good user experience.

ViScan ensure a smooth user experience. Its effectiveness was validated through black-box testing, confirming ease of navigation and clear presentation of detection results, as detailed in Table 1.

TABEL I
VISUALIZATION OF SCORE

No	Feature	Test Case	Expectation Result	Test Result
1	Authentication	User fills out the registration form and presses the “Register” button.	User successfully registers and is directed to the login page.	Valid
2	Authentication	User login with the correct username and password.	User successfully logs in and is directed to the main page (home page display).	Valid
3	Homepage	User presses the floating button on the main page.	The system opens the camera page for skin cancer detection.	Valid
4	Skin Cancer Detection	User takes a picture and the system processes the image.	The system displays image processing indicators and detection results.	Valid
5	Skin Cancer Detection Results	The system displays the detection results page.	The detection results display an image with label accuracy levels.	Valid

V. CONCLUSION

This research successfully developed a smartphone-based application utilizing the YOLOv10x model and Ionic Framework to provide efficient, accurate, and accessible skin cancer screening. The primary goal was to create a user-friendly, AI-powered tool for early detection, designed to work on low-spec mobile devices, particularly for users in underserved or remote areas with limited access to healthcare. The app demonstrates a high level of accuracy, with the YOLOv10x model achieving 84.2% precision, 86.3% recall, 89.2% mAP, and 85.2% F1 Score, making it effective in detecting skin lesions. Its intuitive UI/UX design ensures a seamless user experience, enabling individuals to quickly and easily assess potential skin lesions. The application's effectiveness was further validated through black-box testing, confirming smooth navigation and clarity in presenting detection results. Future developments may incorporate additional features such as medication guides, personalized health recommendations, or offline mode support, further enhancing its usability. The system architecture also holds promise for future upgrades to detect more complex or advanced types of skin cancer, expanding its potential to improve skin health screening globally.

DAFTAR PUSTAKA

[1] A. H. Roky et al., "Overview of skin cancer types and prevalence rates across continents," Cancer Pathog. Ther., vol. 3, no. 2, pp. 89–100, Mar. 2025, doi: 10.1016/j.cpt.2024.08.002.

[2] A. M. Ali, M. S. Dawood, M. K. Taher, and F. A. Zghair, "Preliminary study for non – invasive optical detection of squamous and basal cell carcinomas," Biomed. Eng. OnLine, vol. 11, no. 1, p. 88, Dec. 2012, doi: 10.1186/1475-925X-11-88.

[3] B. Shetty, R. Fernandes, A. P. Rodrigues, R. Chengoden, S. Bhattacharya, and K. Lakshmanna, "Skin lesion classification of dermoscopic images using machine learning and convolutional neural network," Sci. Rep., vol. 12, no. 1, p. 18134, Oct. 2022, doi: 10.1038/s41598-022-22644-9.

[4] A. M. Smak Gregoor et al., "Artificial intelligence in mobile health for skin cancer diagnostics at home (AIM HIGH): a pilot feasibility study," eClinicalMedicine, vol. 60, p. 102019, Jun. 2023, doi: 10.1016/j.eclinm.2023.102019.

[5] F. Rubagumya et al., "Feasibility Study of a Smartphone Application for Detecting Skin Cancers in People With Albinism," JCO Glob. Oncol., no. 6, pp. 1370–1375, Nov. 2020, doi: 10.1200/GO.20.00264.

[6] S. Albahli, "A Robust YOLOv8-Based Framework for Real- Time Melanoma Detection and Segmentation with Multi- Dataset Training," Diagnostics, vol. 15, no. 6, p. 691, Mar. 2025, doi: 10.3390/diagnostics15060691.

[7] Student in Dept of MME, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India et al., "Real-Time Skin Disease Detection and Classification Using YOLOv8 Object Detection for Healthcare Diagnosis," Int. J. Multidiscip. Res. Anal., vol. 07, no. 12, Dec. 2024, doi: 10.47191/ijmra/v7-i12-09.

[8] J. Tan, Y. Chen, and S. Jiao, "Visual Studio Code in Introductory Computer Science Course: An Experience Report," Mar. 10, 2023, arXiv: arXiv:2303.10174. doi: 10.48550/arXiv.2303.10174.

[9] A. A. Smaragdina, R. C. Nur Susilowati, R. Ahmadi, Y. D. Andriansyah, and J. B. Md Yunos, "JSinAja: Javascript Programming Learning Application (React JS, React Native, Node JS) Mobile Based Using Problem Based Learning Method," Lett. Inf. Technol. Educ. LITE, vol. 6, no. 1, p. 5, May 2023, doi: 10.17977/um010v6i22023p5-10.

[10] M. Hort, M. Kechagia, F. Sarro, and M. Harman, "A Survey of Performance Optimization for Mobile Applications," IEEE Trans. Softw. Eng., vol. 48, no. 8, pp. 2879–2904, Aug. 2022, doi: 10.1109/TSE.2021.3071193.

[11] Oscar Danilo Gavilánez Alvarez et al., "Comparative Analysis of Cross-Platform Frameworks," J. Namib. Stud. Hist. Polit. Cult., vol. 33, May 2023, doi: 10.59670/jns.v33i.874.

[12] A. Maulana Komarudin, A. Nurul Huda, and D. Agistira, "PENERAPAN TEKNIK WEB SCRAPING PADA SITUS IMDb DENGAN NODE JS," J. Siliwangi Seri Sains Dan Teknol., vol. 8, no. 2, Dec. 2022, doi: 10.37058/jssainstek.v8i2.6371.

[13] M. N. Jeyaraj, S. Sucharitharathna, C. Senarath, Y. Kanagaraj, and I. Udayakumara, "Cognitive Visual-learning Environment for PostgreSQL".

[14] E. Cengil, "Weld Defect Detection with YOLOv10," NATURENGS MTU J. Eng. Nat. Sci. Mal. Turgut Ozal Univ., Dec. 2024, doi: 10.46572/naturengs.1592956.

[15] B. S. Ali, H. Nasir, A. Khan, M. Ashraf, and S. M. Akbar, "A Machine Learning-Based Model for the Detection of Skin Cancer Using YOLOv10," in 2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia: IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/ICSIPA62061.2024.10686198.

[16] X. Yu, Y. Tan, S. Gao, Y. Zhang, and H. Fang, "Research and application of real-time human posture recognition technology based on TensorFlow.js and CNN model," in International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), Y. Yue, Ed., Wuhan, China: SPIE, Nov. 2024, p. 221. doi: 10.1117/12.3049898.

[17] T. Huang, "FEAD: Figma-Enhanced App Design Framework for Improving UI/UX in Educational App Development," Nov. 22, 2024, arXiv: arXiv:2412.06793. doi: 10.48550/arXiv.2412.06793.

[18] S. Deochake, V. Channapattan, and G. Steelman, "BigBird: Big Data Storage and Analytics at Scale in Hybrid Cloud," Mar. 22, 2022, arXiv: arXiv:2203.11472. doi: 10.48550/arXiv.2203.11472.

[19] T. H. H. Aldhyani, A. Verma, M. H. Al-Adhaileh, and D. Koundal, "Multi-Class Skin Lesion Classification Using a Lightweight Dynamic Kernel Deep-Learning-Based Convolutional Neural Network," Diagnostics, vol. 12, no. 9, p. 2048, Aug. 2022, doi: 10.3390/diagnostics12092048.

[20] A. Newaz and F. S. Haq, "A Novel Hybrid Sampling Framework for Imbalanced Learning," SSRN Electron. J., 2022, doi: 10.2139/ssrn.4200131.

[21] F. Grina, Z. Elouedi, and E. Lefevre, "Learning from Imbalanced Data Using an Evidential Undersampling-Based Ensemble," in Scalable Uncertainty Management, vol. 13562, F. Dupin De Saint-Cyr, M. Öztürk-Escoffier, and N. Potyka, Eds., in Lecture Notes in Computer Science, vol. 13562. , Cham: Springer International Publishing, 2022, pp. 235–248. doi: 10.1007/978-3-031-18843-5_16.

[22] E.-B. C. And Alternative Medicine, "Retracted: An Improved Machine Learning Model for Diagnostic Cancer Recognition Using Artificial Intelligence," Evid. Based Complement. Alternat. Med., vol. 2023, no. 1, p. 9873948, Jan. 2023, doi: 10.1155/2023/9873948.