SpotCheckAI: An Analysis Tool for Suspicious Skin Lesions Utilizing Image Recognition

Rafferty Leung[#1], Henry Wong[#1]

Seidenberg School of Computer Science and Information Systems, Pace University
1 Pace Plaza, New York, NY 10038, USA
[1]rafferty.leung@pace.edu
[2]hwong@pace.edu

Abstract— **Patients requiring time-sensitive treatment for dermatological illnesses may face life-threatening consequences due to the prolonged wait times for dermatology appointments. SpotCheckAI is a web application that employs machine learning for self-diagnosis and triaging in dermatology. The full-stack web application consists of a convolutional neural network (CNN) model for image classification. This paper presents a literature review of some current technologies on the market, their advantages, shortcomings, and why there is a need for more open-source technologies such as SpotCheckAI. SpotCheckAI is a free and accessible application that provides a potential solution for physicians and patients. By providing a user-friendly platform for self-diagnosis and triaging in dermatology, SpotCheckAI offers an accessible and efficient way for patients to address their dermatological needs promptly, potentially preventing life-threatening consequences.**

Keywords— **web development, dermatology, melanoma, image classification, machine learning, neural networks, deep learning**

I. INTRODUCTION

SpotCheckAI is an self-diagnosis application that arose in an effort to help potentially streamline the physician workflow in triaging patents and scheduling patient appointments. Across all sub-specialties, the average wait time for an appointment with a physician is 26 days. [1] For patients with time-sensitive matters, waiting that length of time could result in more prolonged and complicated treatment plan.

SpotCheckAI is a progressive web application (PWA) that can help triage patients when seeing a dermatologist or could potentially help provide preliminary analysis to a patient. The application can be broken down into two major components: the machine-learning model for image classification and the client-side user interface. The model used for this study is a convolutional neural network (CNN) that attempts to correctly classify image inputs from the client-side user interface. Both components make up the web application.

{INDICATE THE OUTLINE HERE}

II. LITERATURE REVIEW

This section can be divided into four distinct parts, which feature an analysis of four organizations that offer comparable tools in the market along with their current methodologies.

A. SkinVision

SkinVision is a subscription-based mobile app that uses artificial intelligence to analyze photos of skin lesions and provide users with an instant risk assessment of whether the lesion appears to be benign or potentially cancerous. Users can take photos of their skin using the app and receive a risk assessment within 30 seconds. [2] SkinVision has tiered subscription services and is currently only available in the European Union.
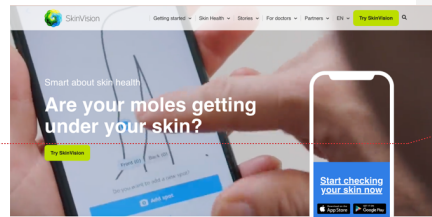

Fig 1. Screenshot of the SkinVision Webpage. [2]

> **Commented [RL1]:** Maybe also integrate the disconnect between ML and medicine. Tools like this bridge the gap

B. SkinScan

SkinScan is a mobile application designed to assist in the early detection of skin cancer. The app uses artificial intelligence and machine learning algorithms to analyze images of skin lesions and moles for signs of potential skin cancer. Users take a photo of their skin lesion or mole with a smartphone camera and upload it to the SkinScan

platform. The app analyzes the image and provides a risk assessment of the lesion, indicating whether it is low, medium, or high risk for skin cancer. [3] SkinScan is also only available in the European Union.
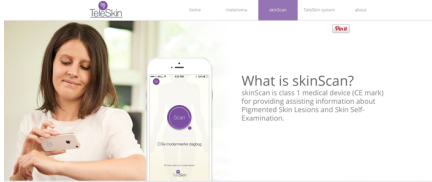

Fig 2. Screenshot of the SkinScan Webpage. [3]

C. UMSkinCheck

UMSkinCheck is a free mobile app developed by the University of Michigan that allows users to perform skin self-examinations and receive instant risk assessments for potential skin cancer. The app guides users through the self-examination process, providing step-by-step instructions on how to take photos of their skin and identifying the areas to focus on. The photos are then analyzed by the app using artificial intelligence and machine learning algorithms to provide an instant risk assessment of the lesions. [4]

Fig 3. Screenshot of the UMSkinCheck webpage. [4]

D. DermAssist

DermAssist is a skin health tool developed by Google Health that uses artificial intelligence to assist users in identifying common skin conditions. The tool allows users to take three photos of the affected area from different angles using their smartphone camera, and then analyzes the photos using machine learning algorithms to provide an instant assessment of the most likely skin condition. [5] DermAssist provides users with information on the condition, including common causes and treatments, and offers suggestions for when to seek medical attention.
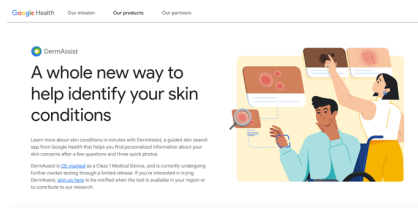

Fig 4. Screenshot of the DermAssist webpage. [5]

III. EVALUATION AND ANALYSIS

A self-diagnostic application is designed to assist individuals in identifying their illness based on their reported symptoms. However, it's important to note that such applications are not completely reliable as they rely on algorithms and previously documented datasets rather than the expertise of a trained medical professional. The purpose of diagnostic applications is to complement a physician's practice and help streamline their patient workflow more efficiently.

In principle, self-diagnostic applications already exist with the explosion in popularity in WebMD and optimized search engines. Users currently can look up potential diseases, treatments, and outcomes based on their symptoms. Users typically will then take this synthesized information with them to their physician's office to gain further understanding and a confirmed diagnosis.

Diagnostic applications have become increasingly popular in recent years, with many users relying on them to help identify potential health concerns based on their symptoms. [6] However, these apps often come with limitations and drawbacks. Moreover, some apps may incorporate advertisements which may draw away from the purpose of providing accurate medical advice.

Here, we analyze and identify the shortcomings of current applications on the market in order to better understand their limitations and improve their functionality. Table 1 displays a high-level

functionality overview of four applications on the market.

Application Name	Platform	Subscription Required	Available for Public Use
SkinVision	Mobile Application	Yes	EU Only
SkinScan	Mobile Application	Yes	EU Only
UMSkinCheck	Mobile Application	No	Yes
DermAssist	Mobile Application	Unknown	Authorized Physicians Only

For the purposes of analysis, SkinVision and SkinScan ("paid mobile applications") will be compared together as they both employ similar methodologies.

The paid mobile applications use a subscription model where users pay, upload, and display a degree of certainty that a lesion is cancerous or benign. No public information was available about the machine learning model or algorithms employed. Furthermore, benchmarking criterion listed on the companies' respective websites are questionable due to study and sample size. [7] Some other issues noted were: small sample size, photos that did not meet evaluation criteria were excluded, no follow up for study participants to see if cancers were identified by physicians but missed by apps. [8] In addition, peer reviewed journals for these two applications noted poor accuracy. [7]

UMSkinCheck's benchmarking is unknown however, it is available for download in the Android and Apple App Store. The University of Michigan released the application back in 2012. At the time of writing, based on user reviews, the application appears to be defunct. [9] In addition, based off of the version history in the Apple App Store, the machine learning model used in this application has not been updated since application inception. Only small updates were made to the user interface and for compatibility with devices and operating systems. [10]

DermAssist is Google's version of a skin lesion classifier. It is not publicly available and little information is known about the application other than users can upload three photos and the program will output the proposed skin condition. DermAssist can identify 90% of commonly searched skin conditions. Other benchmarks regarding DermAssist are unknown. [5]

The four presented applications have similar functionalities where they receive an input and utilizing its internal algorithms, a prediction is made. All options presented are absent of a holistic peer review, namely, missing peer reviewed material regarding the algorithms deployed. The focus has been on benchmarks rather than methodologies. There is a significant need for more open-source and freely available tools such that better detection algorithms can be employed.

The four applications presented employ similar core functionalities. They all take an input and apply internal algorithms to predict the output. However, none of the presented options have undergone peer review regarding itself algorithms but rather their overall benchmarks. This raises concerns regarding efficacy and reliability of the algorithms implemented in these applications.

It is imperative that we invest more in the development of open-source and freely available tools. Such resources will enable us to create better detection algorithms that can tackle broader issues. These tools will allow for a more transparent and collaborative approach to research and development, which can lead to better results and greater confidence in the algorithms used.

The absence of peer reviews in these applications does not necessarily mean that the algorithms and internal methodologies employed are not effective predictors. However, the absence of such a process leaves room for doubt which can potentially have significant ramifications.

Therefore, it is crucial that open-source and freely available tools are available and subject to peer reviews that makes remarks on algorithm use and benchmarks.

IV. METHODOLOGY

SpotCheckAI's feature implementation is similar to websites used today- a frontend user interface and a backend interface containing the data processing.

The goal of SpotCheckAI is to be a web application that attempts to predict suspicious lesions , attempt to stream line a physician's practice, and employ transparency in the machine learning model used as well as its degree of certainty. SpotCheckAI's feature implementation has a frontend user interface and a backend interface containing the machine learning model. The details of the technologies employed as follows:

1. *Front End* – The client-side of SpotCheckAI uses HTML, CSS, JavaScript, Ionic, and React. This enables use of the website with seamless responsive user interface integration across any device with browser access.
2. *Back End* – The server-side of SpotCheckAI is a REST API implemented using Django and Django REST framework to send RESTful requests and receive RESTful responses on the client-side. The machine learning algorithm was implemented using the Keras Library with TensorFlow as the backend engine. More details regarding the implementation of the server-side technologies are discussed below.

A. Django
Django is a robust web-framework written in Python and has gained popularity among developers for its ability to facilitate rapid development of server-side web applications, making it a preferred choice for building complex web projects. [11]

Django was selected for use due to its robustness, large user community, and maturity in the market.

B. Django REST Framework
Django REST Framework (DRF) is a toolkit used to build Web APIs. RF provides a set of reusable tools and building blocks that make it easy to create APIs that can handle complex data types, authentication, and permissions. It supports viewsets and routers for defining Create, Read, Update, Delete (CRUD) operations, and has built-in support for authentication methods. DRF simplifies the process of building Web APIs by utilizing Django. [12]

DRF was selected to build the REST API because of its authentication methods. It allows for secure access control to the machine learning model as well as secure data protection for the uploaded images.

C. Keras and TensorFlow
Keras and TensorFlow are open-source libraries used for building and training machine learning models. TensorFlow provides low-level APIs whereas Keras provides high-level neural network APIs. Keras simplifies the process of building and training machine learning models. [13]

D. Image Classification
The decision to utilize a CNN for image classification was made due to its proven ability to handle such tasks with great accuracy and effectiveness. A CNN has specialized architecture and abilities to automatically learn and extract relevant features from images. [14]

The different layers used in a CNN are:

Input Layer: This involves resizing images to pass them on to subsequent layers for feature extraction.

Convolution Layer: These filters analyze images to identify features and are also utilized for calculating matching feature points during testing.

Pooling Layer: This layer resizes large images by reducing their size, while retaining the most critical information. It accomplishes this by retaining the maximum value from each window and preserving the best fits of each feature within the window.

Rectified Linear Unit Layer (ReLU): This layer replaces every negative number in the pooling

layer with 0. This prevents the learned values of the CNN from becoming stuck near 0 or becoming too large and approaching infinity, thus ensuring mathematical stability.

Fully Connected Layer: The last layer consists of fully connected layers that receive the high-level filtered images and translate them into labelled categories. [14]
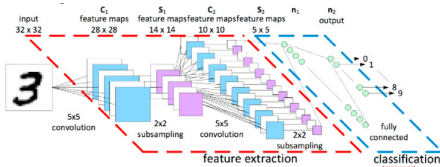

Fig 5. Representation of internal CNN layers. [14]

{Subsequent Sections will be complete by CS692 per Professor Wong}

Currently, self-diagnostic applications use previously documented data sets

One area where both self-diagnostic applications and optimized web searches would fail is when patients have an ultra-rare condition. Patients would have difficulty searching the web for their specific condition and generally speaking, information about their condition may be in the form of technical scientific papers.

A. Text Font of Entire Document

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Recommended font sizes are shown in Table 1.

B. Title and Author Details

Title must be in 24 pt Regular font. Author name must be in 11 pt Regular font. Author affiliation must be in 10 pt Italic. Email address must be in 9 pt Courier Regular font.

TABLE I
FONT SIZES FOR PAPERS

Font Size	Appearance (in Time New Roman or Times)		
	Regular	Bold	Italic
8	table caption (in Small Caps), figure caption, reference item		reference item (partial)
9	author email address (in Courier), cell in a table	abstract body	abstract heading (also in Bold)
10	level-1 heading (in Small Caps), paragraph		level-2 heading, level-3 heading, author affiliation
11	author name		
24	title		

All title and author details must be in single-column format and must be centered.

Every word in a title must be capitalized except for short minor words such as "a", "an", "and", "as", "at", "by", "for", "from", "if", "in", "into", "on", "or", "of", "the", "to", "with".

Author details must not show any professional title (e.g. Managing Director), any academic title (e.g. Dr.) or any membership of any professional organization (e.g. Senior Member IEEE).

To avoid confusion, the family name must be written as the last part of each author name (e.g. John A.K. Smith).

Each affiliation must include, at the very least, the name of the company and the name of the country where the author is based (e.g. Causal Productions Pty Ltd, Australia).

Email address is compulsory for the corresponding author.

C. Section Headings

No more than 3 levels of headings should be used. All headings must be in 10pt font. Every word in a heading must be capitalized except for short minor words as listed in Section III-B.

1) Level-1 Heading: A level-1 heading must be in Small Caps, centered and numbered using uppercase Roman numerals. For example, see heading "III. Page Style" of this document. The two level-1 headings which must not be numbered are "Acknowledgment" and "References".

Commented [RL2]: Talk about what CNN layer was used and why

2) Level-2 Heading: A level-2 heading must be in Italic, left-justified and numbered using an uppercase alphabetic letter followed by a period. For example, see heading "C. Section Headings" above.

3) Level-3 Heading: A level-3 heading must be indented, in Italic and numbered with an Arabic numeral followed by a right parenthesis. The level-3 heading must end with a colon. The body of the level-3 section immediately follows the level-3 heading in the same paragraph. For example, this paragraph begins with a level-3 heading.

D. Figures and Tables

Figures and tables must be centered in the column. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page.

Graphics may be full color. All colors will be retained on the CDROM. Graphics must not use stipple fill patterns because they may not be reproduced properly. Please use only *SOLID FILL* colors which contrast well both on screen and on a black-and-white hardcopy, as shown in Fig. 1.
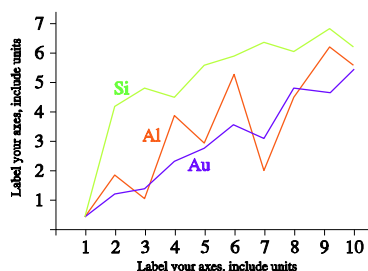

Fig. 1 A sample line graph using colors which contrast well both on screen and on a black-and-white hardcopy

Fig. 2 shows an example of a low-resolution image which would not be acceptable, whereas Fig. 3 shows an example of an image with adequate resolution. Check that the resolution is adequate to reveal the important detail in the figure.

Please check all figures in your paper both on screen and on a black-and-white hardcopy. When you check your paper on a black-and-white hardcopy, please ensure that:
- the colors used in each figure contrast well,
- the image used in each figure is clear,
- all text labels in each figure are legible.

E. Figure Captions

Figures must be numbered using Arabic numerals. Figure captions must be in 8 pt Regular font. Captions of a single line (e.g. Fig. 2) must be centered whereas multi-line captions must be justified (e.g. Fig. 1). Captions with figure numbers must be placed after their associated figures, as shown in   Fig. 1.

Fig. 2 Example of an unacceptable low-resolution image

Fig. 3 Example of an image with acceptable resolution

F. Table Captions

Tables must be numbered using uppercase Roman numerals. Table captions must be centred and in 8 pt Regular font with Small Caps. Every word in a table caption must be capitalized except for short minor words as listed in Section III-B. Captions with table numbers must be placed before their associated tables, as shown in Table 1.

I. References

The heading of the References section must not be numbered. All reference items must be in 8 pt font. Please use Regular and Italic styles to distinguish different fields as shown in the References section. Number the reference items consecutively in square brackets (e.g. [1]).

When referring to a reference item, please simply use the reference number, as in [2]. Do not use "Ref. [3]" or "Reference [3]" except at the beginning of a sentence, e.g. "Reference [3] shows …". Multiple references are each numbered with separate brackets (e.g. [2], [3], [4]–[6]).

Examples of reference items of different categories shown in the References section include:

- example of a book in [1]
- example of a book in a series in [2]
- example of a journal article in [3]
- example of a conference paper in [4]
- example of a patent in [5]
- example of a website in [6]
- example of a web page in [7]
- example of a databook as a manual in [8]
- example of a datasheet in [9]
- example of a master's thesis in [10]
- example of a technical report in [11]
- example of a standard in [12]

V. Conclusions

References

[1] PatientEngagementHIT, "Average Patient Appointment Wait Time Is 26 Days in 2022," *PatientEngagementHIT*, Sep. 15, 2022. https://patientengagementhit.com/news/average-patient-appointment-wait-time-is-26-days-in-2022 (accessed Feb. 21, 2023).

[2] "SkinVision | Skin Cancer Melanoma Detection App | SkinVision," Feb. 24, 2021. https://www.skinvision.com/ (accessed Feb. 21, 2023).

[3] T. doo Bokeljska 7 and 11000 Belgrade, "TeleSkin advanced dermoscopy platform and mobile app for early detection of skin cancer and melanoma," *TeleSkin - high-tech, venture*

backed startup.
https://teleskin.org/skinscan.html (accessed Feb. 21, 2023).

[4] "UMSkinCheck | Michigan Medicine."
https://www.uofmhealth.org/patient%20and%2
0visitor%20guide/my-skin-check-app (accessed Feb. 21, 2023).

[5] "Identify Skin Conditions with DermAssist - Google Health."
https://health.google/consumers/dermassist/ (accessed Feb. 21, 2023).

[6] A. De, A. Sarda, S. Gupta, and S. Das, "Use of Artificial Intelligence in Dermatology," *Indian J. Dermatol.*, vol. 65, no. 5, pp. 352–357, 2020, doi: 10.4103/ijd.IJD_418_20.

[7] K. Freeman *et al.*, "Algorithm based smartphone apps to assess risk of skin cancer in adults: systematic review of diagnostic accuracy studies," *BMJ*, vol. 368, p. m127, Feb. 2020, doi: 10.1136/bmj.m127.

[8] "Some smartphone apps not accurate enough to spot all skin cancers | BMJ."
https://www.bmj.com/company/newsroom/sma rtphone-apps-not-accurate-enough-to-spot-all-skin-cancers/ (accessed Feb. 21, 2023).

[9] "UMSkinCheck - Apps on Google Play."
https://play.google.com/store/apps/details?id=e du.umich.MySkinCheck&hl=en_US&gl=US (accessed Feb. 21, 2023).

[10] "UMSkinCheck," *App Store*, Oct. 03, 2017.
https://apps.apple.com/us/app/umskincheck/id5 22498604 (accessed Feb. 21, 2023).

[11] "Django," *Django Project*.
https://www.djangoproject.com/ (accessed Feb. 22, 2023).

[12] "Home - Django REST framework."
https://www.django-rest-framework.org/ (accessed Feb. 22, 2023).

[13] K. Team, "Keras documentation: About Keras." https://keras.io/about/ (accessed Feb. 22, 2023).

[14] N. Sharma, V. Jain, and A. Mishra, "An Analysis Of Convolutional Neural Networks For Image Classification," *Procedia Comput. Sci.*, vol. 132, pp. 377–384, Jan. 2018, doi: 10.1016/j.procs.2018.05.198.