The Use of Artificial Intelligence for Melanoma Detection

Submitted by

Omar Nayfeh

Computer Science

To

The Honors College

Oakland University

In partial fulfillment of the

requirement to graduate from

The Honors College

Mentor: Hua Ming, Professor of Computer Science

Co-Mentor: Prof Alao, PhD Candidate, Computer Science

Department of Computer Science

Oakland University

December 1st, 2023

Source code for this project can be found at https://github.com/nayfeho/CSI4999.

Abstract

With an increase in the uses of Artificial Intelligence (AI) and Machine Learning (ML), the possibilities of computer science in other fields are endless. Therefore, having developed an application in Python (Programming Language), that uses AI and ML to detect if an image of a skin spot uploaded by a patient or doctor is or is not potentially melanoma with percentage certainty can aid in the early detection phase and save valuable time. Namely, Convolutional Neural Networks (CNN) techniques have been implemented through the help of TensorFlow and Keras, open-source frameworks that aid the development of CNNs, since they are the standard for distinguishing the classification of an image. Through the application's diagnosis, this means that doctors can reevaluate what the best course of action should be to save their patient's life. This project provides a better understanding of how these Artificial Intelligence applications can be used within the medical field. This thesis will also aid others in the same area of expertise to build upon their own applications through this project as a reference.

Current Research

Cancer has been prevalent throughout history, and it has been proven that early detection increases the rate of treating it correctly. Skin cancer is the most common cancer in the United States where 20% of Americans will develop skin cancer in their lifetimes, and "skin cancer affects more people in the United States each year than all other cancers combined [1]". To combat this issue, doctors must detect cancerous skin spots early and accurately, since "Melanoma is the skin cancer with the worst prognosis. If diagnosed early, it can be treated successfully with surgical procedures [1]". However, doctors may misdiagnose whether a patient's skin spot may be cancerous or not due to unfamiliarity or lack of substantial evidence when a special case is involved. Through these misdiagnoses, valuable time is lost that could have otherwise been used to reduce the spread of the cancer to other parts of the body and save the patient's life.

Thus, much research has been conducted to "discuss the fundamentals of ML [machine learning] and its potential in assisting the diagnosis of skin cancer [1]". Das highlighted the uses of Machine learning using convolutional neural networks, but did not go into detail about the development of such applications. While others such as Tschandl, gathered the results to train "a particular type of convolutional neural network (CNN), on the training dataset of a publicly available image benchmark of pigmented lesions containing seven diagnostic categories [2]". This research gave more insight on multiple types of skin cancer where melanoma was taken into account. However, they mostly focused on the results of how accurate the application was using confidence intervals. There is also research "to identify and group the different types of AI-based [Artificial Intelligence] technologies used to detect and classify skin cancer [3]". This illustrated how each technology compared to each other, described the benefits of each technique, and had inconclusive findings on which type of AI- technology is more accurate.

Das further discusses their findings on how accurate their applications were when running it on new images while showing the “usefulness of AI in diagnosis and management of skin cancer [1]”. Other research explains “Image-based AI of individual skin lesions has been shown to both exceed and improve expert dermatologist performance in artificial experimental settings” [4]. Highlighting the necessity of Artificial Intelligence in the diagnosis of melanoma. Also, the resources currently available in other countries across the globe are not sufficient for proper diagnosis. Melakodre mentions “While early diagnosis is paramount for an effective cure for cancer, the current process requires the involvement of skin cancer specialists, which makes it an expensive procedure and not easily available and affordable in developing countries.” [5]. Since AI applications can be accessed easily online or as an application downloaded on the computer itself, those countries would stand to benefit. Yet another study discovered that “Ultimately, the development and validation of AI technologies, their approval by regulatory agencies, and widespread adoption by dermatologists and other clinicians may enhance patient care” [7].

Therefore, having developed an application using Artificial Intelligence and particularly Machine Learning with Convolutional Neural Networks techniques would further benefit doctors and patients in the early detection phase. Since, there have been studies indicating “deep learning convolutional neural networks (CNNs) show great potential for melanoma diagnosis.” [6]. As a result, this could significantly reduce the chances of misdiagnosis because “ skin disease diagnosis is mostly based on visual perception, computer vision algorithms may be able to recognize skin lesions based on their morphology [1]”, which would eliminate any human error.

Aims and Objectives

Introduction

Having developed an application using Artificial Intelligence and Machine Learning using Convolutional Neural Networks (CNN), TensorFlow and Keras, open-source frameworks that aid the development of CNNs, in Python (Programming Language), this application determines whether or not a photo of a skin spot uploaded by a patient or doctor is melanoma with percentage certainty. Since the leading cancer in the United States is skin cancer, it is especially important for patients to have early detection and correct diagnosis of their skin spots, such that they will be given appropriate treatment.

Aims

1. To determine if an uploaded image of a skin spot is or is not melanoma.

2. To calculate the percentage accuracy of the diagnosis.

Objectives

1. Determining if an uploaded image of a skin spot is or is not melanoma influences the course of action doctor's must take when treating the patient based on the diagnosis given.

2. Calculating the percentage accuracy of the diagnosis establishes the level of certainty the application has when determining the diagnosis of the uploaded skin spot.

Methodology

Introduction

This application was developed using Artificial Intelligence and Machine Learning using Convolutional Neural Networks (CNN), TensorFlow and Keras, open-source frameworks that aid the development of CNNs, in Python (Programming Language). The intended usage of the application is to upload a photo of a skin spot, using the appropriate criteria, on a desktop application named MelaKnowma. The application then decides whether the input was good or not, if not the application notifies the user. Otherwise, the application will display the predicted diagnosis of skin spot alongside the percentage certainty. From that point on, the user may either upload another image or quit the application.

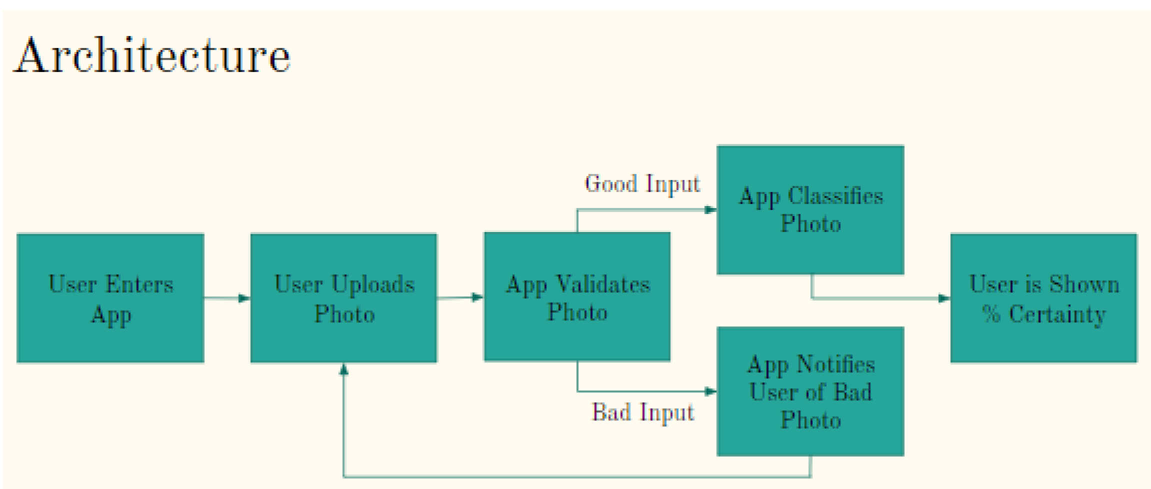

Figure 1: Application's Architectural Design

Backend

First, the primary focus was researching common implementations of image detection using Artificial Intelligence and Machine Learning through various trusted resources online. Doing this determined that CNNs (Convolutional Neural Networks) are the standard for distinguishing the classification of an image between two subcategories, typically developed in Python. Since Python was used, we also needed an appropriate user interface that could be used for the photo upload functionality and to display the diagnosis and percentage certainty. After researching more, it was determined that a Python GUI (Graphical User Interface) using Tkinter, Python Library. This was due to the ease of communication between the backend and frontend, and it was an easy to use API (application programming interface) that was also easy to understand and learn. Another aspect researched early on was the type of dataset needed to be used. Ultimately, a large supervised dataset was chosen to train on such that the model can learn on specifically labeled images to mitigate any sort of misdiagnosis on training images.

When we had begun developing this application, a dataset needed to be provided for the application to train on first. A large supervised dataset with images that have been confirmed to be melanoma, as well as benign skin spots allowed the application to recognize patterns that pertain to each category. Alternatively, an unsupervised dataset could have been used that contains images of unlabeled skin spots, however, that potentially risked the result of false classifications since the application will try to find patterns within all the images instead of two sub-categories. Therefore, this data was collected through trusted resources and datasets found online such as the HAM10000, which contains 10,000 images of confirmed Melanoma skin spots. The figures below are training images of skin spots of each category labeled appropriately for the dataset.

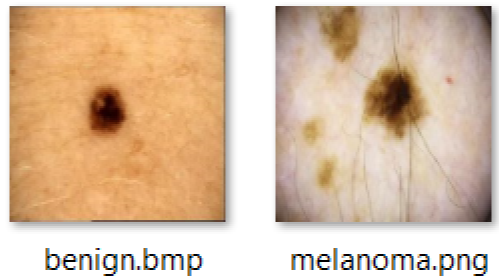

Figure 2: Appropriately labeled training images of skin spots for the dataset

Next, the application preprocessed the training data to ensure unwanted images that do not fit the criteria of our training dataset were removed. This was accomplished by looping through all of the images in our supervised dataset saved in a designated folder named data and authenticating whether or not the images are saved under the file extensions, .jpeg, .jpg, .bmp, and .png. If an image does not have one of these extensions, the image is simply not taken into account when training the model. Following that, all of the images were scaled down such that the quantity of pixels increased, which allowed them to be analyzed more effectively.

Afterwards, the data was partitioned such that to determine how much of the data is getting worked on. The data was split randomly into three different categories for the model to use, and any duplicate images were skipped. The training category contained 70% of all the images in the dataset to use for training the model. The evaluate category contained 20% of images in the dataset to evaluate. Lastly, 10% of the dataset was saved under the test category, such that the model can be tested against itself and determine its accuracy.

Next, Artificial Intelligence (AI) and Machine Learning (ML) were implemented in Python using Convolutional Neural Networks (CNN) techniques to recognize patterns in the model. This was implemented through TensorFlow and Keras since they are free open-source libraries that aid in the development and provide an interface for neural networks. This was accomplished by using pip install on the command line to install the necessary libraries on the local machine and then using an import statement within the data model file, which allows all of the libraries methods to be accessible. A CNN was chosen instead of a Deep Neural Network (DNN) because CNN's are a specialized type of DNN's and the standard for distinguishing the classification of an image between two subcategories, especially with the use of medical analysis because many patterns can be easily recognized. In the case of melanoma, there are four major signs that can be easily detected through the use of a CNN. They include asymmetry where one half of the mole does not match the other, the edges of the mole are irregular, ragged, notched or blurred, the color of the mole is typically darker with the shades of brown or black, and finally the diameter of the mole is larger than six millimeter across. Therefore, the CNN model trained to find these patterns and observations within the dataset given to it.

The architecture of CNN's are feed forward neural networks through the use of filters and pooling, as seen in the figure below.

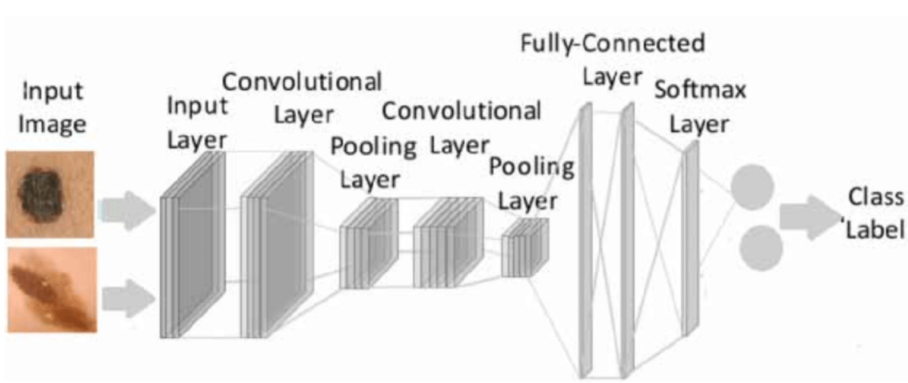

Figure 3: Convolutional Neural Network Layers for the Application

As a result, a sequential model was chosen due to the fact that it uses multiple layers that have exactly one input and output, and they are known for their accuracy and quickness when given limited inputs. These layers also feed forward and process the data sequentially which better suits the architecture used for Convolutional Neural Networks. The implemented CNN has a total of five layers. Initially, the input layer took the image data that was fed into the model. Typically, an image is represented as a 3D tensor, with dimensions corresponding to its height, width, and color channels, which was the supervised dataset in this case. Then, the use of three convolutional layers applied convolutional filters to the input data where each filter detects specific features in the image, like edges or textures. These layers were followed by ReLU (Rectified Linear Unit) activation functions to introduce non-linearities. Next, three pooling layers were implemented in between each convolutional layer to down-sample the spatial dimensions (height and width) of the input data. Particularly, a common technique called max pooling was used, where the maximum value in a patch of the input data is taken. Additionally, three dense layers, also known as fully connected layers, were used to flatten the data into a 1D vector and learn global patterns prevalent within the data. The last dense layer has as many neurons as there are classes to predict, and it uses a softmax activation function to output probabilities for each class. Finally, the output layer results in a graph that illustrates a line graph of the accuracy of the model against the value accuracy given by the training dataset, as seen in the figure below.

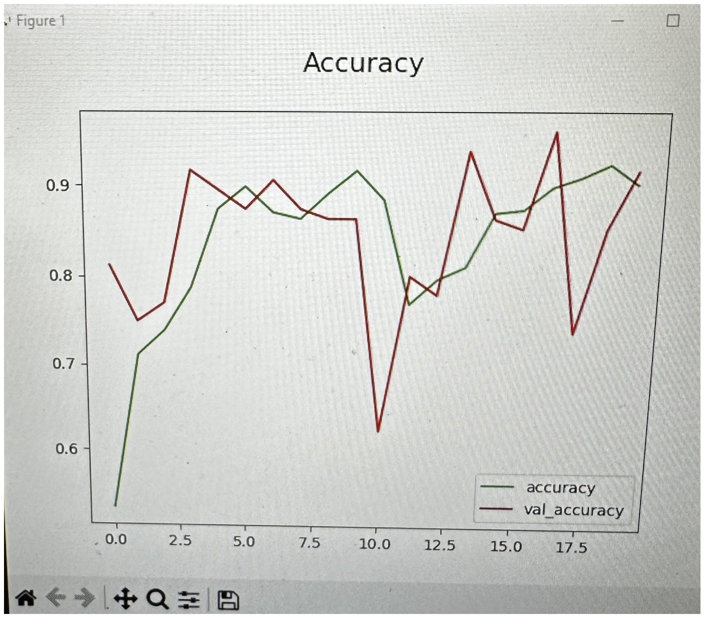

Figure 4: Graph of the accuracy of the model against the accuracy given by the training dataset

It is worth mentioning that as the amount of training data increases, the accuracy of the model will also increase. After the CNN model was trained, new data could also be evaluated. A new image would have to be provided to be used in the model.predict method, which results in a diagnosis and its percentage certainty rounded to second decimal. Therefore, this model was saved as a model version 1.0 so that whenever new images are added to the training data, an updated model can also be saved. Finally, a loadmodel method was created such that it can be called upon at any time in the frontend.

Frontend

For the user interface portion of the application, a Python GUI (Graphical User Interface) using the Python Library Tkinter was used. The GUI was developed to be very user friendly. The GUI's size is 600 by 400 pixels, which contains the application's name, MelaKnowma, a select photo button, a path entry input, an identify button, and an example of proper criteria for the image that will be uploaded. The criteria states that the image must be a close up of a skin spot with adequate lighting. The application was developed such that once the user clicks on the select photo button, they can navigate to the desired image if its file extension is one of the following: .jpeg, .jpg, .png, .bmp, as seen in the figure below. Alternatively, the user may also drag and drop the desired image into the GUI, or they can type in the entire path to the file. Then, the user may press the identify button, which will communicate with the backend.

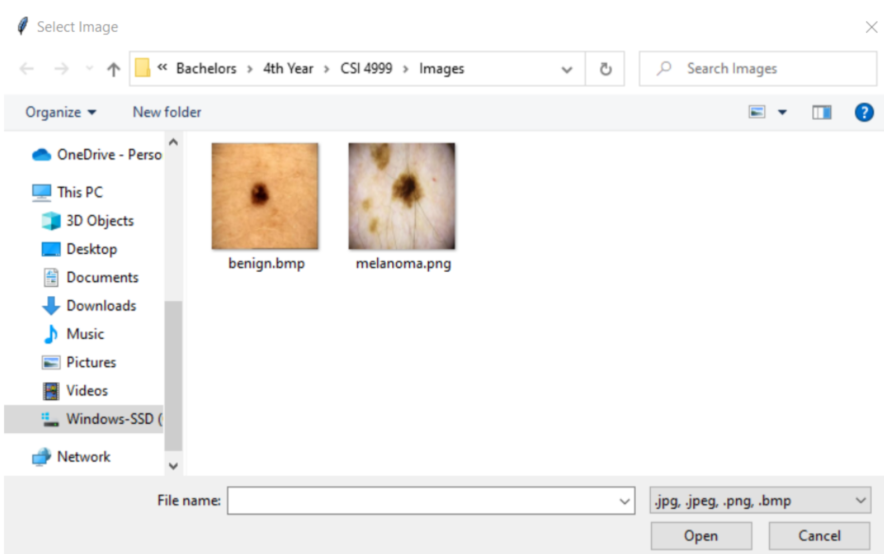

Figure 5 : Selection of skin spot image

After the image is uploaded and scaled down, a verification method is used to verify that the image is in fact an image of a skin spot in the proper criteria as mentioned earlier. If these conditions are not met, an error message will display indicating such. The user will have to upload another image. Once the image is accepted, the loadmodel function from the backend is called to determine its diagnosis in a quick and prompt manner. The resulting screen shows the predicted diagnosis, its percentage certainty, and the uploaded image to authenticate to the user that they selected that image in case of a misclick. The user was finally prompted to upload another image or quit the application.

IRB

This project does not involve human subject research and does not require IRB approval.

Discussions

The outcome of this project was an application that detects if an image of a skin spot uploaded is or is not potentially melanoma with percentage certainty.The intended usage of the application is to upload a photo of a skin spot, using the appropriate criteria, on a desktop application named MelaKnowma. The application then decides whether the input was good or not, if not the application notifies the user. Otherwise, the application will display the predicted diagnosis of skin spot alongside the percentage certainty. From that point on, the user may either upload another image or quit the application.

Through this application, patients and doctors alike can have the diagnosis of the skin spot delivered quickly and accurately, such that valuable time can be saved. Proper treatment would also be given early on since the chances of misdiagnosis is much lower. It was determined that this application had an accuracy of 80% on all diagnoses, which is substantially better than the current accuracy of 70% in the medical field as of now. It is also worth noting that if the model was given more confirmed melanoma and benign skin spot images, the accuracy could potentially be much larger.

This application can potentially allow other researchers and developers insight on the development of Artificial Intelligence and Machine Learning applications that are not only for Melanoma detection but also the medical field as a whole. Other developers may build upon the application's source code to improve the performance and quickness of the diagnosis.

Conclusions

Personally, this project has allowed me to learn more about Artificial Intelligence and Machine Learning developed in Python, since I have not gotten the chance to familiarize myself with it yet. This is due to scheduling conflicts for the classes pertaining to those topics. Therefore I felt it was necessary to develop a project with these technologies, as well as, the different learning techniques like Convolutional Neural Networks through the help of open source frameworks such as TensorFlow and Keras. I also learned much more about Graphic User Interfaces (GUIs) in Python with the use of Tkinter, a Python library used for the development of GUIs.

Furthermore, I also gained experience in the development of a large scale project with a team of four other students. I learned more about Agile Scrum methodology and how it was actually put into practice, through weekly scrum meetings, a scrum board that contained all of our sprints and user stories that needed to be developed. This entire experience has enhanced my skill set and will directly correlate with future experiences because there is an increasing demand of Software Engineers for Artificial Intelligence and Machine Learning applications.

Biographical Note

As a student majoring in Computer Science, I aspire to further my technical knowledge by developing relevant and practical applications that can benefit society. I also plan to pursue a master's degree in Computer Science to learn new technologies and build on prior principles, as well as, a master's degree in Business Administration such that I can lead like minded individuals who also possess great technical skills. Eventually, becoming a manager in software development is my goal such that I can manage revolutionizing software products, so that they can be built both more efficiently and quickly, while also benefiting the end user. This project has allowed me to collaborate with a team of four other students within several sprints with the common goal of developing an application that detects the chances of melanoma from a skin spot image. In particular, I was a developer who was tasked with developing parts of the melanoma detection model and the user interface, as well as, collecting images for the dataset. Therefore, I learned more about project development, communication, and technical skills. Specifically, sprint goals and managing priority of certain criteria, how to work effectively with a team that has differing schedules and advanced Python with the use of Artificial Intelligence and Machine Learning. As a result, I gained practical experience and skills that will all later translate into my higher education and profession.

Thank You

Throughout the course of my undergraduate degree in Computer Science, I have truly enjoyed my time at Oakland University. I would like to thank all of the professors I have taken courses with for their sheer dedication and willingness to teach and aid every student. I have increased not only my technical knowledge but also my respect for their mission of developing the minds of tomorrow. Specifically, I would like to acknowledge the support of my Faculty Mentors, Dr. Ming and Miss Alao, for their expert suggestions and constructive criticism on my Honors College Thesis and Proposal. I would also like to thank my thesis advisor, Dr. Beckwith, for all of her guidance and continued support throughout the process of this Thesis. Last but not least, I would like to thank my family and friends for all of their encouragement and support in my undergraduate studies. They have made a difference in my life and continue to support me in earning your degree. I would not be the person I am today without the help of all these influential people. So from the bottom of my heart, thank you!

Bibliography

Citation Style: IEEE Citation

[1] K. Das et al., “Machine learning and its application in skin cancer,” International Journal of Environmental Research and Public Health, vol. 18, no. 24, p. 13409, 2021. doi:10.3390/ijerph182413409

[2] P. Tschandl et al., “Human–computer collaboration for Skin cancer recognition,” Nature Medicine, vol. 26, no. 8, pp. 1229–1234, 2020. doi:10.1038/s41591-020-0942-0

[3] A. Takiddin, J. Schneider, Y. Yang, A. Abd-Alrazaq, and M. Househ, “Artificial Intelligence for Skin Cancer Detection: Scoping Review,” Journal of Medical Internet Research, vol. 23, no. 11, 2021. doi:10.2196/22934

[4] M. A. Marchetti et al., “Prospective validation of dermoscopy-based open-source artificial intelligence for melanoma diagnosis (prove-AI study),” npj Digital Medicine, vol. 6, no. 1, 2023. doi:10.1038/s41746-023-00872-1

[5] N. Melarkode, K. Srinivasan, S. M. Qaisar, and P. Plawiak, “Ai-powered diagnosis of Skin cancer: A contemporary review, open challenges and future research directions,” *Cancers*, vol. 15, no. 4, p. 1183, 2023. doi:10.3390/cancers15041183

[6] J. K. Winkler *et al.*, “Melanoma recognition by a deep learning convolutional neural network—performance in different melanoma subtypes and localisations,” *European Journal of Cancer*, vol. 127, pp. 21–29, 2020. doi:10.1016/j.ejca.2019.11.020

[7] E. J. Beltrami et al., “Artificial Intelligence in the detection of skin cancer,” Journal of the American Academy of Dermatology, vol. 87, no. 6, pp. 1336–1342, 2022. doi:10.1016/j.jaad.2022.08.028