Artificial Intelligence

Smart Mirror for Skin Disease Detection

By:

Haya Abuteen - 20190126
Aws Burmawi - 20190470
Sami Alamin - 20190753

Instructor:

Dr. Rawan Ghnemat

I. INTRODUCTION

Smart home applications' main goal is to make your life easier in most possible ways. One of the most important aspects of life is health, integrating smart home applications with health applications to improve one's health and make it easier to deal with can be a very effective idea. This research paper focuses on dermatology, skin diseases, and the ability to classify them using machine learning, and image processing techniques.

Imagine a scenario where a mirror has the remarkable capability to identify skin diseases as soon as you wake up or while you prepare for your daily activities. The potential of such a smart home mirror becomes evident when considering the possibility of waking up one morning and discovering the presence of contagious skin disease. In this situation, the smart mirror could play a crucial role in detecting skin disease, thereby simplifying the management and containment of the contagious condition. Armed with the knowledge of being infected with a contagious disease, individuals would be empowered to take appropriate actions to safeguard those around them and effectively address the disease. This revolutionary application of smart home mirrors holds immense promise in ensuring the early detection, containment, and appropriate response to contagious skin diseases, thus contributing to public health and well-being.

II. BACKGROUND AND MOTIVATION

Over the years, due to multiple reasons such as excessive use of hygiene products that contain a lot of unknown chemicals and how those chemicals affect the skin, skin diseases became more present, leading to people not paying the right amount of attention to some of those diseases which caused them not knowing what type of skin disease themselves suffered from, if that disease is infectious, and how badly it impacts the skin. There is a high need for a sustainable, efficient at-home solution for these problems to make the effect of skin diseases less dangerous for the community.

The motivation behind this project is to be able to identify the most common skin disease that one might suffer from while at home as fast as possible and efficiently, to be able to deal with it the proper way, ensuring high protection for others living in a community when one suffers from an infectious skin disease.

III. GOAL AND OBJECTIVES

Our overarching aim is to transform the identification of common skin diseases by utilizing everyday household equipment in an innovative and accessible way. We want to empower individuals to take control of their skin health through user-friendly methods integrated into their daily lives, ultimately enabling early detection and timely intervention for improved outcomes.

Utilize mirrors' many useful features to help people diagnose skin conditions by allowing them to check their own skin and take high-resolution pictures for examination.

Analyze the collected skin photos using cutting-edge artificial intelligence and sophisticated image processing architectures and algorithms to quickly and effectively identify potential skin illnesses.

A. PREVIOUS WORK

1) Dermnet Dataset with CNN and Average Pooling CNN (Convolutional Neural Network) is a powerful algorithm commonly used to analyze visual data, like images. It uses convolutional layers that work like filters to extract important features, much like how our brains focus on specific aspects when we look at something. This allows the CNN to understand and interpret visual data in a meaningful way.
   With average pooling, a sliding window calculates the average value of the components inside the window as a downsampling technique in CNNs. It condenses features, decreases spatial dimensions, and aids in preventing overfitting.
   The way we structure the sequential model, including the choice and arrangement of layers like convolutional, pooling, and fully connected layers, can indirectly influence accuracy. By carefully designing the model, we enhance its capacity to learn and extract important features from the data, ultimately improving its ability to capture underlying patterns. This leads to better accuracy as the model effectively represents the information within the data.
   In this case, the average pooling retains the essence of the features by smoothing the image, which results in down-sampling and removing any noise present in the picture. All of this resulted in an accuracy that reached 72%.
2) Dermnet Dataset with CNN and Inception Model Similar to the work above, CNN was used, however, the inception model was used instead of average pooling. The inception module plays a crucial role in capturing features at different scales within a neural network. By simultaneously implementing various convolutional and pooling operations at each layer, it enables the network to learn a diverse range of features. The concatenated output from the inception module is then passed to the subsequent layer for further processing. This module consists of a max pooling layer and a set of convolutional filters (1x1, 3x3, and 5x5) that all operate on the same input simultaneously. The final output of the inception module is obtained by depth-wise concatenation of the output feature maps from each operation which resulted in a 40% accuracy.
3) Dermnet Dataset with CNN and Max Pooling using VGG16 Architecture In this case, they used a technique called max pooling in a CNN model known as VGG16. Max pooling is like looking at a picture and finding the most noticeable features. The VGG16 architecture is quite powerful, with 16 layers that include convolutional and fully connected layers. It has been widely

used and achieved great results in tasks like classifying images and detecting objects. Its simplicity and deep structure have made it a popular choice for transfer learning, where pre-trained VGG16 models are used as a starting point for training on different datasets or tasks. Max pooling is an important part of the VGG16 architecture. It helps reduce the size of the image, makes sure the model recognizes features regardless of their position and captures the most important details. This helps the model build a hierarchy of visual features and learn more complex patterns. Hence why this model had an accuracy of 69.97%

4) Dermnet Dataset with CNN and ANN An Artificial Neural Network (ANN) is a computational model inspired by the human brain. It's made up of interconnected units, called artificial neurons, that work together to process and transmit information. ANNs are good at handling large amounts of data and can be used for tasks like recognizing patterns, making classifications, or even understanding language and images. Overall, an ANN is a powerful tool that mimics how our brains process information, and it has applications in many areas of artificial intelligence and machine learning.

   When it comes to detecting skin diseases, using a combination of CNNs and ANNs can be quite effective.

   CNNs are great at understanding images and picking up on important patterns. They can learn to recognize specific features associated with different skin conditions.

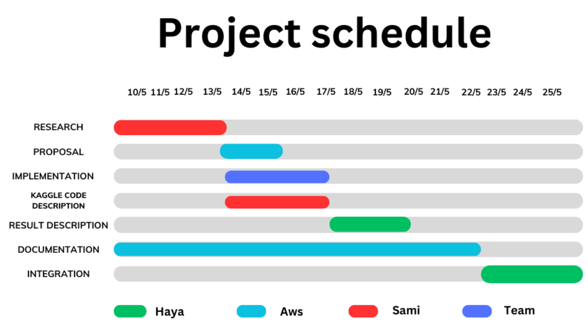
   In this combined approach, the CNN acts as a feature extractor. It learns from a large set of labeled skin images and captures important visual characteristics.

   The output from the CNN is then fed into an ANN, which acts as a classifier. The ANN takes the learned features and makes predictions about the presence or absence of specific skin diseases.

   By combining the strengths of CNNs and ANNs, we can benefit from the CNN's ability to extract relevant features from skin images and the ANN's decision-making capabilities.

   This combination of CNN and ANN has shown promising results in skin disease detection and other medical image analysis tasks. It allows us to improve accuracy and reliability by utilizing the strengths of both types of neural networks. Hence why the accuracy here was 87.93% which was the highest.

IV. PROJECT SCHEDULE



V. IMPLEMENTATION

A. TRIALS

With respect to all models and methodologies prior to implementing the CNN inception model, we have tried using different architectures and models. For the first try, we used RESNET50 architecture, we ran into multiple problems while implementing that solution. The model was slow, and gave low accuracy values. The model was trained on 100 epochs and gave a high training accuracy, but the model was overfitting it produced 27% testing accuracy. To solve that problem, more layers were added, which gave a higher testing accuracy but the model was still overfit. The second try gave less promising accuracy values, we combined three different models CNN,ANN, and VGG16. We concluded that the combined model was too complex, it gave us very low training and testing accuracy values.

B. IMPLEMENTATION DESCRIPTION

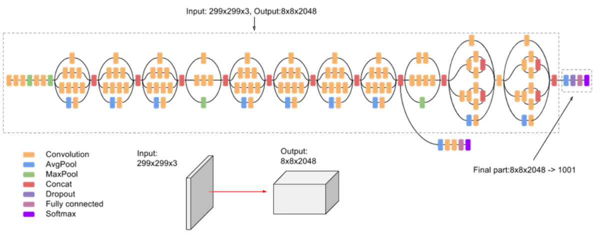
The code has been restructured to be more organized and easier to read. Instead of having everything mixed together, the code now groups imports together, clearly indicating which libraries are being used. Additionally, variables and directories are defined at the beginning, allowing for easy locating and modification when necessary.

A notable improvement is the implementation of data augmentation techniques. Data augmentation helps to diversify the training data by randomly applying transformations like rotation, zoom, and flipping to the images. This can enhance the model's robustness and ability to handle variations in real-world data.

Furthermore, the code explicitly sets the image size and batch size, providing a clear understanding of the input data dimensions. This makes it simpler to comprehend and modify the code according to specific requirements.

Another enhancement is the utilization of the tf.keras.preprocessing.image-dataset-from-directory function to load the training and testing datasets. This function simplifies the process by automatically organizing the data based on the directory structure. It saves time and effort by automatically assigning labels to the images.

Moreover, the code explicitly defines the loss function, optimizer, and metrics, making it easier to understand and modify these parameters according to different scenarios or



preferences.

Both versions of the code employ the InceptionV3 architecture. However, in the original code, the base-model is initialized without pre-trained weights, whereas the updated code initializes it with pre-trained weights by default unless specified otherwise.

To summarize, the updated code showcases a more organized and readable structure. It incorporates data augmentation for improved model performance and provides clearer control over important parameters. These enhancements contribute to better comprehension, usability, and flexibility for individuals working with the code.

C. EXPECTED RESULTS

The main objective of the updated code is to train an image classification model using transfer learning and the InceptionV3 architecture. By leveraging transfer learning, we aim to achieve better accuracy and lower loss as the model learns to classify images from the given dataset. However, it's important to note that the actual performance of the model may vary based on several factors. These factors include the quality of the dataset, the number of classes, the duration of the training, and the complexity of the images within the dataset. Each of these elements can have an impact on how well the model performs and how effectively it can classify the images.

D. ACTUAL RESULTS

The model was overfit, it had a training accuracy of 89% and a testing accuracy of 39%. This is due to the complexity of the model used, which is the inception model. The advantages of this model are low information loss and efficient and effective image-processing tasks. However, some noticeable weakness it its complexity which results in the overfitting problem that occurred.

VI. CONCLUSION

One important side observed during working on the code was the significance of computational power; it is as important as the efficiency of the methodology used, which lead to using TPU and GPU instead of using the CPU alone which gave faster and more efficient results,

An idea that the team working on this project agreed on was, adding a healthy skin folder to the dataset would make the idea of a smart mirror detecting skin diseases much more executable in the real world and would give more coherent results.

Last but not least, the data used in any Artificial intelligence project is the most important part, and it is what defines a great artificial intelligence application from a better one.

REFERENCES

1) Abdalgwad, M. (2023, May 8). CNN+Ann. Kaggle. https://www.kaggle.com/code/mohamedabdalgwad/cnn-ann
2) Arivan, T. (2022, November 30). Skin disease classification. Kaggle. https://www.kaggle.com/code/tarivan/skin-disease-classification
3) Manudeep123. (2022, June 5). 23 skin-diseases images classification(40% ACC). Kaggle. https://www.kaggle.com/code/manudeep123/23-skin-diseases-images-classification-40-acc
4) Papers with code - inception-V3 explained. Explained | Papers With Code. (n.d.). https://paperswithcode.com/method/inception-v3
5) Sanrai. (2023, February 7). Demoproject. Kaggle. https://www.kaggle.com/code/sanrai/demoproject

APPENDIX. CODE IMPLEMENTATION

https://www.kaggle.com/hayaabuteen/ai-project-implementation