<u>A proposal of deep-learning-based Magic Mirror modules to identify specific health aspects.</u>

By: Yash Gupta, Atreya Rawat, Collin Li

Abstract

One of the most prevalent healthcare problems in the world today is that people choose not to visit the doctor's office regularly. Even when people show symptoms of sickness, they decide not to go for unsubstantial reasons, including not having enough time. Our proposed solution is to use the technique of deep learning and the advancement in IoT infrastructure to help make healthcare more accessible to the general public. We used a device called the Magic Mirror, which has software capable of displaying a plethora of data. This data includes weather, calendar, news, etc. Another fundamental aspect of the Magic Mirror is the customization feature, which allows a user to display whatever they want on the screen. We used this capability to create two widgets capable of detecting eyebags and acne on a human face. Each of these modules was created used convolutional neural networks. We had three approaches to creating an eye bag module, and we separated the full face images into four sections. We found that by cropping full faces images into just the eye region of the face, the eye bag module performed best. We also found that cropping photos into the chin region of the face optimized the acne module's performance.

Introduction

Due to the prevalent resources in medicine, the number of bacterial diseases prevalent in society has been steadily decreasing. However, there are still some major issues that almost every human being goes through in their lifetime. These include acne and sleep problems. However, a plethora of people avoids seeking medical care even when they suspect it may be necessary. Even individuals who are experiencing multiple symptoms choose not to go to the doctor. Graver, the reasons that people have for avoiding medical care are nuanced and highly varied.

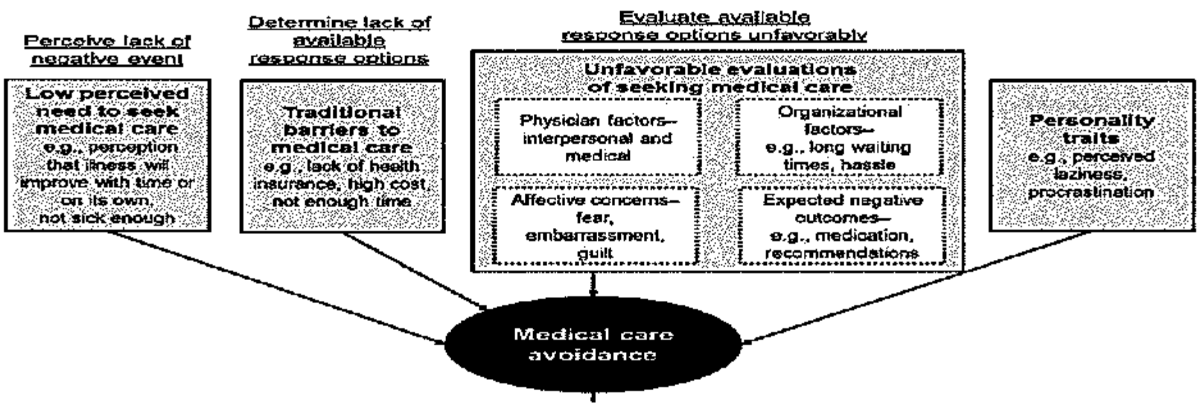


Figure 1: Depiction of the excuses humans make toward not wanting to receive proper medical care. The leading reasons include not having enough time and believing there is no need.

Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4351276/

Researchers have invented techniques to help doctors keep track of their patients outside of their facilities. Such devices include Fitbit, implantables, and smart clothing. However, these devices are not capable of identifying some of the most prevalent health problems associated with Americans today: acne vulgaris and insomnia. Approximately 15 million Americans are classified as having severe acne with over 50 million being associated with acne symptoms in general. Additionally, approximately 60 million Americans are affected by sleep disorders such as insomnia. While some scientists at Microsoft and Nestle Skin Health have created a module capable of identifying the severity of acne for a human, the module is not easily understandable by the general citizen, essentially making the module worthless. Our solution comprises of using the technique of deep learning and an IoT device called a Magic Mirror to help make modules more accessible for the general public. The modules we created were capable of identifying acne and eye bags.

IoT Infrastructure

As the world is becoming more digitized, there is an exponential increase in the amount of data that people generate. To help sort through this pool of data, the concept of IoT infrastructure was invented. Internet of Things (IoT) describes a wide ecosystem where devices are interconnected through the Internet, allowing them to readily exchange and process data. This concept has and will continue to revolutionize almost every industry in the world, such as agriculture, retail, and transportation. The device we used to help improve the healthcare industry is called the Magic Mirror, a raspberry pi powered monitor behind a double-sided mirror.

Figure 2: Picture of the Magic Mirror, an IoT device, which we will upload our two modules onto
Source: Student generated image

We chose to use this device because it has an associated black web-page with a unique feature of customization. In other words, this web page allows users to choose the modules they want displaying in their mirror. This aspect of customization allowed us to create our own widgets that are compatible with the

Magic Mirror webpage. The modules we created returned results that can be sent directly to doctors and health physicians to better assist patients.

Deep Learning

Since the advent of the 21st century, computer scientists have been researching ways to create a true artificial intelligence system. In other words, computer scientists are trying to create a machine capable of thinking and processing information just like humans, only in a way that is faster and more advanced than the human mind. One step towards a true A.I system has been the creation of deep learning, a computer technique located under the branch of machine learning. Machine learning is the idea that computers are capable of performing specific tasks without being given any instruction and can only rely on patterns and inferences. Deep-learning helps computers recognize and utilize patterns and inferences among data. The way that this is accomplished is through the use of artificial neural networks.

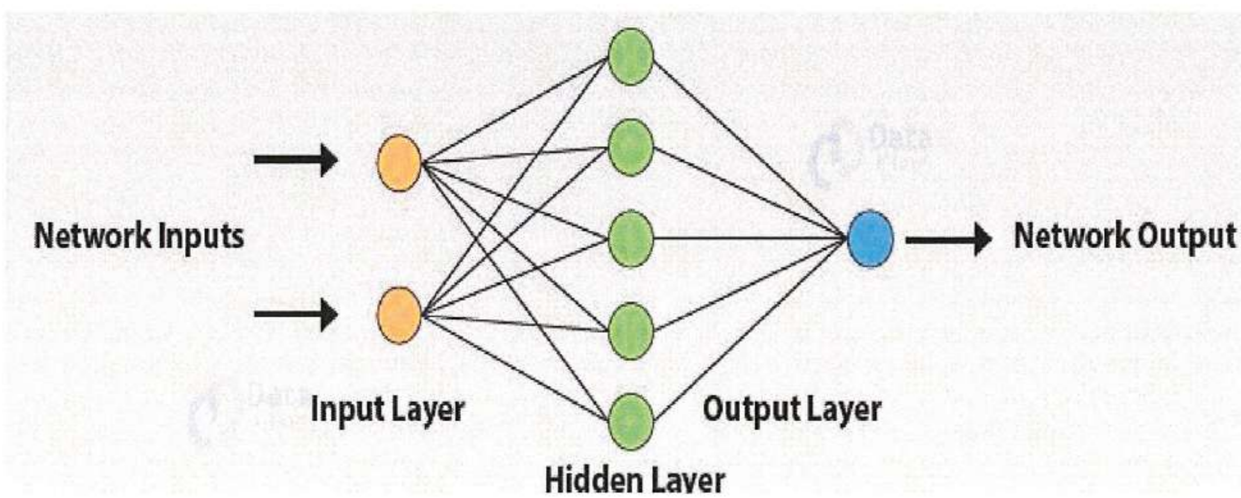


Figure 3 shows the general layout of every artificial neural network.
Source: https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning/

Every neural network has three types of layers: an input layer, a hidden layer, and output layer. The input layer consists of neurons that each hold a value between 0 and 1. Typically, each neuron is used to represent a pixel of a picture. The value each neuron receives from the pixel is determined by how active the pixel is. In other words, the higher the RGB value of a pixel, the higher the value of a neuron will be.

$$z = x_1 * w_1 + x_2 * w_2 + \ldots\ldots + x_n * w_n + b * 1$$
$$\hat{y} = a_{out} = sigmoid(z)$$
$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

Figure 4: This figure shows the equations that neural networks use to determine the activations of each consecutive neuron
Source: http://neuralnetworksanddeeplearning.com/chap6.html

Each neuron in the input layer is used to determine the value for one neuron in the next sequential layer. Hidden layers are used to determine what features are contained in an image. However, these values are altered through the use of weights and biases. Weights represent the importance of each input neuron to a neuron in the first hidden layer. Each neuron has a different weight from the other neurons in the same layer and different weights depending on which neuron in the second layer it is corresponding to. This procedure repeats until the network reaches the output layer. At this point, the network simply returns the neuron with the highest value.

However, we used convolutional neural networks(CNN) in our project to achieve more accurate results due to spatial structure in images. The way that this network works is by grouping together pixels in a picture to create what is called a *local receptive field.* These pixels will represent one neuron in the first hidden layer. This grouping is then shifted left, right, up and down, to create the rest of the neurons in the first hidden layer. The reason for using this is to identify certain structures that may appear frequently within entire images.

Hypothesis'

Acne: The acne module created will help determine the severity of acne vulgaris for a human being.

Eyebags: The eye bag module will return a certainty percentage of approximately 85%

Materials

- Databases of images for each different module
- Raspberry pi 4 Model B
- Magic Mirror Monitor
- HD USB camera

Methodology

For both modules, we first had to gather images that would be used to train and test the deep learning models. We gathered the images for each using the same method which was to look up images on google. For eye bags, we looked up images of people with eyebags and people without eyebags and put the images into 2 different files which were eyebags and no eye bags. The same was done for acne where we looked up images in google of people with acne and people without acne and put those images into 2 files as well which were acne and no acne.

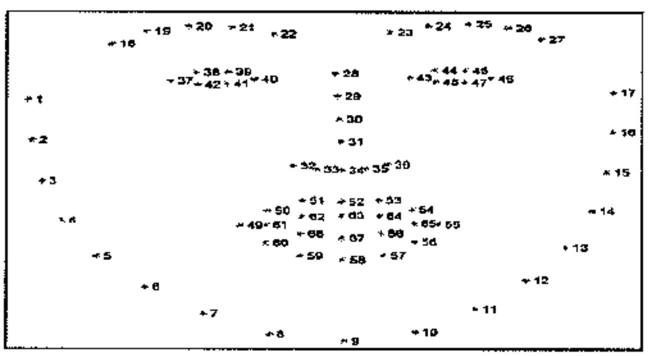


Figure 5: the 68 landmark points that dlib facial landmarks plots/identifies https://medium.com/@aanilkayy/extract-eyes-in-face-i%CC%87mages-with-dlib-and-face-landmark-points-c45ef480c1

Another similarity is that we used the facial recognition landmark command to crop our images that we gathered from the internet for the eyebags and acne. What this command does is that after locating faces in an image, it identifies specific points on a face. These locations include the Eyes, Eyebrows, Nose, Mouth, and Jawline.

Also, for the acne and eyebags dataset, we had to crop each image so that they would be the same number of pixels because you cannot train a model with a different number of pixels. We did this by creating code from Python to make each image be a certain number of pixels. We did this for both the acne and the eyebags images.

To train both of these modules, we imported the Keras deep learning library from TensorFlow, which is a neural network library that is written in Python. Keras is the deep learning neural network API. Using TensorFlow, we imported different files within that since TensorFlow has several different libraries. For both the eyebags and acne code, we used a pickle file to store the images from the datasets. A pickle file is used for converting a python object into a character stream to store in a file. We decided to use this pickle file because it is very efficient in the way that we do not have to add extra code and it can also create graphs by storing recursive functions. Also, we

```
"""
Created on Sat Jan 11 16:36:46 2020

@author:
"""


import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D

import pickle
import matplotlib.pyplot as plt

from keras.utils import to_categorical

X = pickle.load(open("X.pickle", "rb"))
y = pickle.load(open("y.pickle", "rb"))




X = X/255.0


model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape = X.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dropout(0.25))


model.add(Flatten())
model.add(Dense(128))
model.add(Activation("relu"))

model.add(Dense(128))
model.add(Activation("relu"))


  model.add(Dense(3))

model.add(Dense(3))
model.add(Activation("softmax"))


model.compile(loss="sparse_categorical_crossentropy",
              optimizer="adam",
              metrics=["accuracy"])





history = model.fit(X, y, batch_size=32, epochs=55, validation_split=0.1)


model_json = model.to_json()
with open("model.json", "w") as json_file :
    json_file.write(model_json)

model.save_weights("model.h5")
print("Saved model to disk")

model.save("model")
```

Figure 6: Using Adam Optimizer to compile a model
Source: Student generated code

used the softmax function to output the probability that someone has eyebags or acne. We also used Adam optimizer to train each of our models. The benefits of using an Adam optimizer are that it is very efficient, it is very easy to optimize and it is very easy to get started with so we did not need to know a lot of information about this, and it does not require a lot of memory. Also, in the code for the acne and eye bags, we made 2 categories which were acne and no acne for the acne category and for the eyebags category, we made eyebags and no eye bags.

1. Eye Bag Module

The first module attempted was to detect the symptom periorbital puffiness also known more commonly as eye bags. Along with creating a deep-learning-based module, we started off by using non-deep learning methods to compare two images by using feature detection from the python library cv2. This would allow us to first get a better understanding of how normal image comparison worked and then be able to show the differences in performance from deep-learning modules. We had two deep-learning models for the Eye Bag module. For both models, we first started off by collecting a wide range of photos sorted into two categories (Eyebag or No Eyebag). We had to resize the images into the same number of pixels in order for the deep learning model to process the images correctly. The resulting sizes were images that were 75 by 150 pixels.

Figure 7: Showcases some of the images we used to train the first and second models of our program
Source:
https://www.pinterest.es/pin/322993222997198l5/

For the first model, we did not crop out any specific sections of the images. For the second model, we decided to crop out only the eye-to-nose region of the face. We decided to attempt both approaches because we believed that in the full face images, the model could potentially be looking for other factors such as hair to identify if a subject has eye bags or not. We wanted to determine if limiting these extra variables made a difference within the accuracies. Finally, we trained the two models using the process described above. Both used a batch size of 32 images, with Model 1 resulting in 17 layers and 25 epochs and Model 2 resulting in 9 layers and 15 epochs.

2. Acne Module

Our second module attempted on classifying Acne vulgaris also known more commonly as acne or pimples, which is one of the most common skin conditions in the world. We started off deciding the split the face into four distance sections for better predictions. This is due to the fact that if we trained a model with full faces, any sign of acne could trigger the model to predict a result of mild/severe acne. Thus we decided to split the dataset into four categories: forehead, left cheek, right cheek, and chin due to the wide range of areas these sections cover over the face.

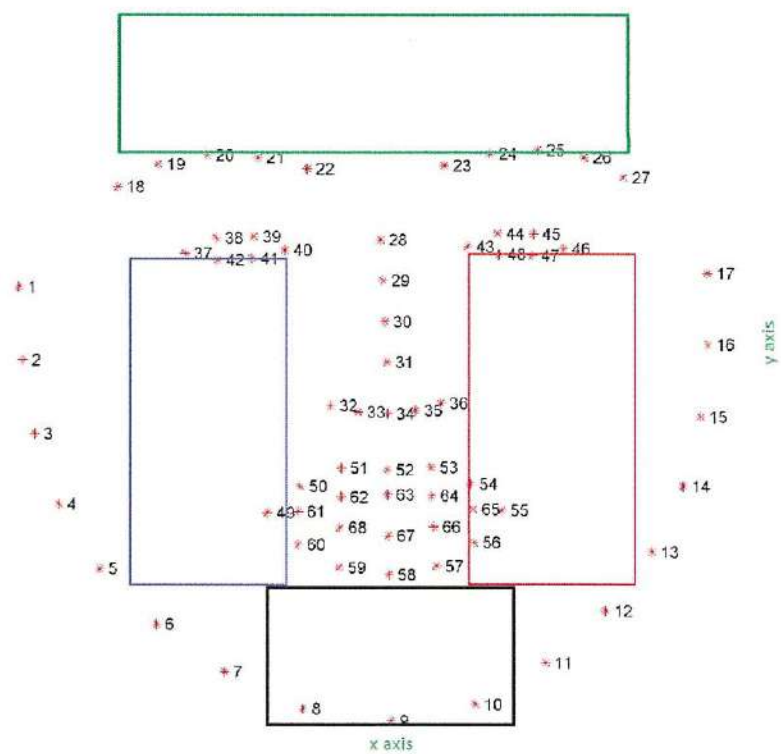


Figure 8: Draws out how the different regions of the face were cropped based on the 68 landmarks
Source: Student made figure

It is also due to the fact that these areas of skin have the most oil (sebaceous) glands and thus have the greatest chance acne appearing there. We approached this cropping problem the same as in eyebags as we used dlib's facial detection and more specifically with ageitgey's python facial detection repository on Github. The way we cropped the four different sections are shown in figure 8. As shown in the figure, we cropped the sections with rectangles. For the left cheek and right cheek, we used the furthest x-coordinate eyebrow point (which are points 18 and 27) from the nose for the outer side of the rectangle (to crop) and closest x-coordinate eyepoint (which are points 40 and 46) from the nose for the inner side of the rectangle (to crop). The y-coordinates were based on the lowest eyepoint (which are points 42 and 47) and the lowest bottom lip point (point 58). The chin was based on entirely points from the mouth and the lower chin. We used points 49 and 55 as side boundaries and points 58 and 9 as height boundaries. Finally, the forehead was a bit more complicated as dlib's 68 point facial landmarks didn't map out the hairline. Luckily, we could use the golden ratio of a human's face shown in figure 9 to determine the upper part of the rectangle (to crop). The sides for the forehead cropping were based on the eyebrow points 18 and 27 while the bottom of the rectangle (to crop) was based on the point highest on the face (in this case it would be point 25). To train the models, we used 4 layers with an optimizer of 'adam' and the sparse_categorical_crossentropy loss to compile our model. We used categorical_crossentropy instead of binary_crossentropy since we also wanted to identify the severity of the acne. We used a batch size of 32 and an epoch size of 20 for all the four models trained for the four parts of the

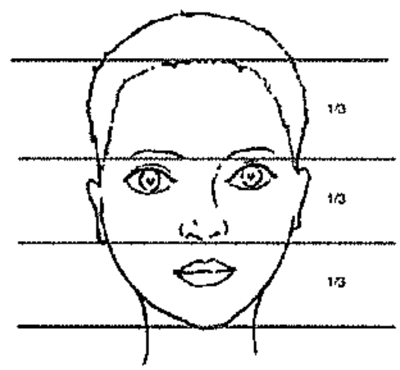


Figure 9: Demonstrates the ⅓ ratio of the face for cropping the forehead for acne https://www.sciencedirect.com/science/article/pii/S1808869418303161

face. Finally, we resized all the different sections: Chin by 30x30, Forehead by 30x30, Left Cheek by 30x30, and finally Right Cheek by 30x30. These variables were used due to the fact that they gave the best results after training and testing the model.

Results & Discussion

1. Eye Bag Results

For the non-deep learning approach that we attempted for the eye bag module, the program returned whether or not an eye bag was detected. However, these results were extremely inaccurate. This was apparent because, in photos with clearly identifiable eyebags, the program would predict that the subject had no eye bags. These inaccuracies demonstrate that by not using deep learning, the creation of successful healthcare modules will never exist.

Figures 10 + 11: These two figures show the results for Model 1 of the Eyebag module.
Source for figure 10 and 11: collected and modified from https://www.pinterest.com/pin/647533252642877694/

The results for the model we trained with full face images are shown above. For each of the training photos, the module outputs a percentage of how accurate it believes its answer is and shows that percentage on top of the photo. When the model detects an eye bag, the writing appears red, and for when no eye bag is detected, the writing will appear green. As shown in figures 10 + 11, the first model we created was inaccurate when tested with cropped and uncropped images. For example, in figure 10, the fourth person in the first row clearly has eyebags but is classified as having no eyebags. In figure 11, the first person in the first row is said to have eyebags yet she clearly does not have any eye bags. The reason for these huge inaccuracies is that there are multiple variables being considered when the deep learning model creates its weights and biases. One factor includes hair.

For Model 2, our results were more accurate than the results for the non-deep learning approach and Model 1. For example, as shown in figure 12, the model is now classifying more images into the correct categories. In other words, the computer is more accurate in determining if there are eye bags with higher percentages as well. However, the model is still not perfect. Additionally, a trend we noticed was that people with darker skin tones were classified as having eyebags more often than others. Due to this we looked back at the images and did not find this correlation. This trend revealed a problem with Model 2 of the eye bag module.

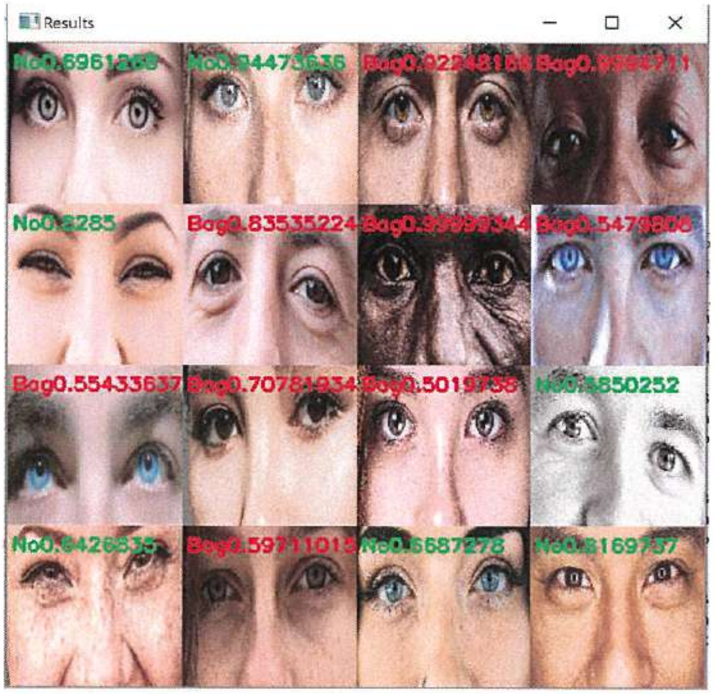


Figure 12: This figure shows the results for Model 2 of the Eyebag module.
Source for figure 12: collected and modified from https://www.pinterest.com/pin/647533252642877694/

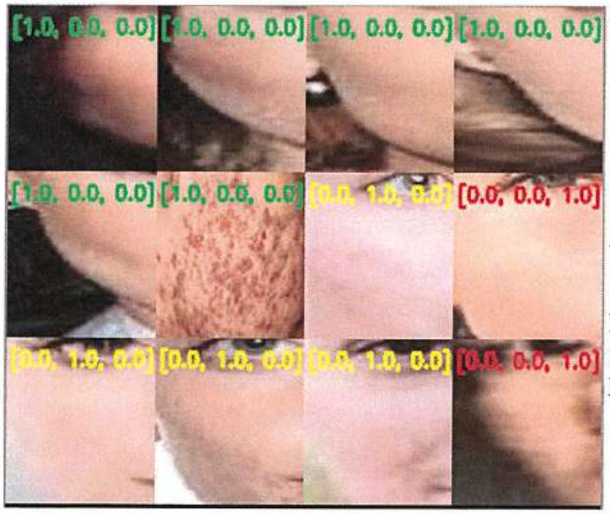
The validation accuracy that Model 1 returned was below 40 %. This demonstrates that using full-face images is not ideal for real-world applications and the validation accuracy of Model 2 was 64.71%. So while the results fared better when using cropped images, there still may be unknown variables that are affecting the accuracy of Model 2. Based on these results, we can observe that Model 2 was more effective for creating an eye bag module.

2. Acne Results

Our acne modules for the chin, forehead, left cheek, and right cheek, have been improved and retrained as we sought to find the best rescaling sizes, number of layers, types of layers, and many more variables. However, due to our limiting dataset and time, are models are nowhere near perfect. The figures below give a visual representation of our testing accuracy. Red symbolizes severe acne, yellow symbolizes mild acne, and green symbolizes clear/no acne. The descriptions of the figures will also tell the accuracy of the model based on the dataset used to train ("acc"), the accuracy on photos based on images never before seen by the model ("val_acc").

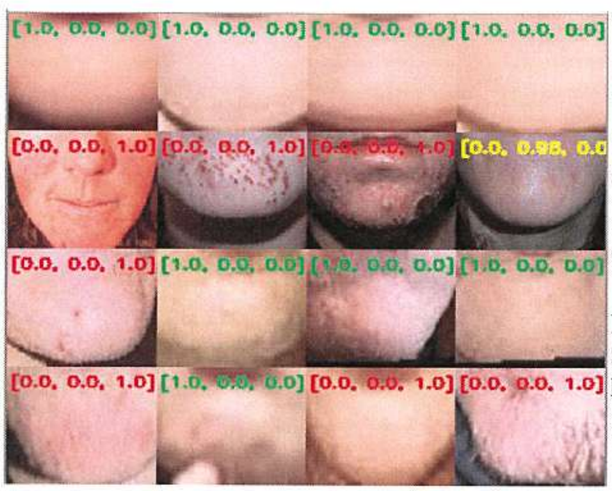
The Forehead model was trained on 208 images and then validated on 11 images. The model returned an accuracy of 89.71% and a validation accuracy of 63.64%.

Figure 13: Showcases the forehead model results
Source: Collected and modified from
https://dermnetnz.org/topics/acne-face-images/



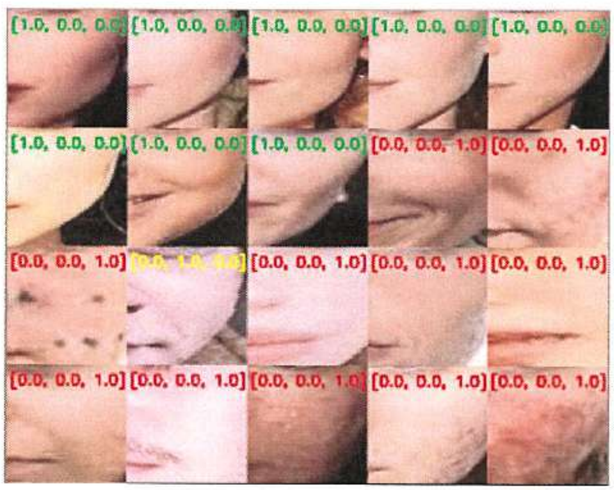
The Right Check model was trained on 216 images and then validated on 12 images. The model returned an accuracy of 93.08% and a validation accuracy of 78.58%.

Figure 14: Showcases the right cheek model results
Source: Collected and modified from
https://dermnetnz.org/topics/acne-face-images/



The Chin model was trained on 216 images and then validated on 12 images. The model returned an accuracy of 86.57% and a validation accuracy of 75.00%.

Figure 15: Showcases the chin model results
Source: Collected and modified from
https://dermnetnz.org/topics/acne-face-images/



The Left Check model was trained on 240 images and then validated on 13 images. The model returned an accuracy of 92.92% and a validation accuracy of 84.62%.

Figure 16: Showcases the left cheek model results
Source: Collected and modified from
https://dermnetnz.org/topics/acne-face-images/

The Chin model has the worst training accuracy of 86.57% most likely due to the smaller dataset used to train. On the other hand, the forehead model has the worst validation accuracy most likely due to the fact the interference of hair in various photos.

Conclusion

Deep learning allows computers to recognize patterns and inferences in large amounts of data and then utilize them to solve problems. Both an acne detector and an eye bag scanner were successfully created. However, the accuracy is still not 100% which may cause concern among the general public. This may be due to the fact that the databases of images we were using were extremely inadequate for creating a deep-learning model. Most models that are trained with up to 4000 images are considered limited so our database is not sufficient enough. The lack of pictures may also be the cause of not having many hidden layers throughout our deep-learning model.

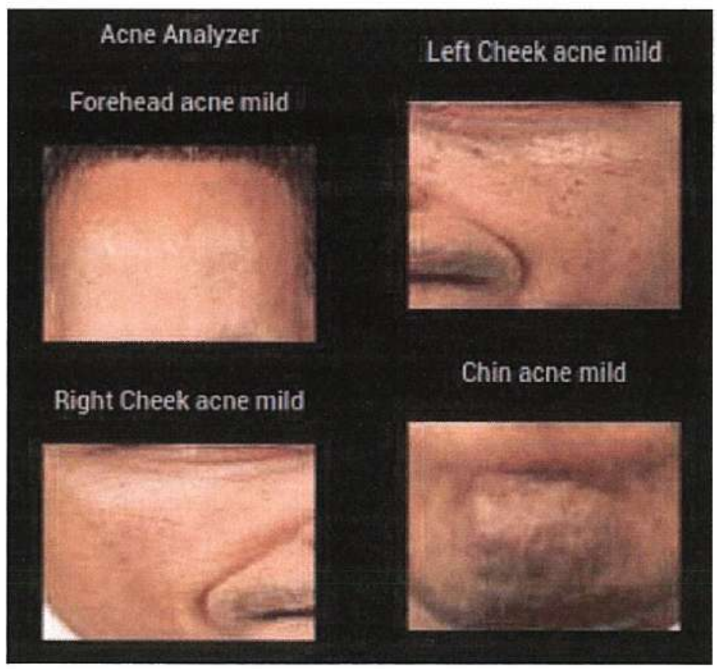


Figure 17 + 18: Figure 17 depicts how the eye bag module would look on a Magic Mirror monitor. 18 demonstrates how the acne module would divide a subjects face into four sections and gives its prediction above the sections. The four section would be displayed vertically Source: Student generated images.

Future Research

Increasing the number of images included in our databases will help increase the accuracy of the two modules. We would also increase the number of pooling layers that our convolutional neural networks contain. By implementing these two changes within our research, the accuracy of our results should be improved. Additionally, to properly diagnose insomnia, a variety of symptoms need to be detected along with the presence of eye bags. For example, insomnia has been correlated to an increase in body weight. So creating a module that helps detect BMI(Body-Mass-Index) may better help doctors in diagnosing insomnia to their patients. There is also a multitude of other health issues prevalent throughout the world. One example of this is cardiovascular disease, which nearly 120 million Americans deal with every year. Some new modules that would help doctors diagnose this disease could be a heart-rate scanner, cholesterol level determiner, and many more. Instead of using deep-learning, perhaps the broader technique of machine learning may be more beneficial in revolutionizing the healthcare industry. Finally, another aspect of this research that we can improve upon including more IoT devices so that we can track these health symptoms throughout the day and in more than one location. This work shows the promise of using deep-learning and the Magic Mirror device to create new healthcare modules. The amount and accuracy of these modules will only increase as the world becomes ever more digitized and the prospect of a true artificial intelligence system becomes a reality.

References

1. Alhammad, S. A. (2018). Face Detection for Pulse Rate Measurement. 2018 1st International Conference on Computer Applications & Information Security (ICCAIS). doi: 10.1109/cais.2018.8442034
2. Ballinger, Brandon. *DeepHeart: Semi-Supervised Sequence Learning for Cardiovascular Risk Prediction.*
3. Boyko, N., Basystiuk, O., & Shakhovska, N. (2018). Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). doi: 10.1109/dsmp.2018.8478556
4. Colantonio, S., Coppini, G., Germanese, D., Giorgi, D., Magrini, M., Marraccini, P., … Salvetti, O. (2015). A smart mirror to promote a healthy lifestyle. Biosystems Engineering, 138, 33–43. doi: 10.1016/j.biosystemseng.2015.06.008
5. Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), 1(1), 886–893. doi: 10.1109/cvpr.2005.177
6. "Facial Acne Images." *Facial Acne Images | DermNet NZ*, 2007, dermnetnz.org/topics/acne-face-images/.
7. Firouzi, Farshad, et al. "Internet-of-Things and Big Data for Smarter Healthcare: From Device to Architecture, Applications and Analytics." *Future Generation Computer Systems*, North-Holland, 23 Sept. 2017, www.sciencedirect.com/science/article/pii/S0167739X17319726.
8. G, Swapna, et al, "Automated Detection of Cardiac Arrhythmia Using Deep Learning Techniques." *Procedia Computer Science*, Elsevier, 8 June 2018, www.sciencedirect.com/science/article/pii/S187705091830766X.
9. Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep Convolutional Neural Networks with transfer learning for computer vision-based

data-driven pavement distress detection. Construction and Building Materials, 157, 322–330. doi: 10.1016/j.conbuildmat.2017.09.110

10. Hossain, M. Shamim, and Ghulam Muhammad. "Cloud-Assisted Industrial Internet of Things (IIoT) – Enabled Framework for Health Monitoring." *Computer Networks*, Elsevier, 4 Feb. 2016, www.sciencedirect.com/science/article/abs/pii/S1389128616300019.
11. Kapoor, Ajay. "Deep Learning vs. Machine Learning: A Simple Explanation." hackernoon.com/deep-learning-vs-machine-learning-a-simple-explanation-47405b3eef08
12. Lakshmi N M, Chandana M S, Ishwarya P, Nagarur Meena, Rajendra R Patil. (2018). IoT based Smart Mirror using Raspberry Pi, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCESC – 2018 (6)13 doi: N/A
13. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. doi: 10.1038/nature14539
14. Meola, Andrew. "The Critical Role of Infrastructure in the Internet of Things." *Business Insider*, Business Insider, 21 Dec. 2016, www.businessinsider.com/internet-of-things-infrastructure-architecture-management-2016-10.
15. Microsoft. "Microsoft/Nestle-Acne-Assessment." *GitHub*, github.com/microsoft/nestle-acne-assessment.
16. Nielsen, and Michael A. "Neural Networks and Deep Learning." *Neural Networks and Deep Learning*, Determination Press, 1 Jan. 1970, neuralnetworksanddeeplearning.com/chap6.html.
17. Pathak A., Mishra A., Sarate R., Bhavsar S, Patel N. (2018). Smart Mirror using Raspberry Pi. International Journal of Recent Trends in Engineering and Research, 4(3), 353–358. doi: 10.23883/ijrter.2018.4140.mow8w
18. Poh, M.-Z., Mcduff, D., & Picard, R. (2011). A medical mirror for non-contact health monitoring. ACM SIGGRAPH 2011 Emerging Technologies on - SIGGRAPH 11. doi: 10.1145/2048259.2048261

19. Radha, Mustafa, et al. "Sleep Stage Classification from Heart-Rate Variability Using Long Short-Term Memory Neural Networks." *Nature News*, Nature Publishing Group, 2 Oct. 2019, www.nature.com/articles/s41598-019-49703-y.

20. Rathore, M. Mazhar, et al. "IoT-Based Big Data: From Smart City towards Next Generation Super City Planning." *International Journal on Semantic Web and Information Systems (IJSWIS)*, IGI Global, 1 Jan. 2017, www.igi-global.com/article/iot-based-big-data/172421.

21. Shin, Hoo-Chang, et al. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics, and Transfer Learning." *Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning - IEEE Journals & Magazine,* 2016, ieeexplore.ieee.org/abstract/document/7404017.

22. Sisal, P., Satpute, N., & Pawar, P. V. E. (2018). Smart Mirror Using Raspberry Pi. International Journal of Engineering and Techniques, 4(2), 554-557. doi: 10.29126/23951303

23. Solutions, NH Learning. "Transitioning to the Internet of Things (IoT) - Are You Ready?" *Transitioning to the Internet of Things (IoT) - Are You Ready?* blog.nhlearningsolutions.com/blog/transitioning-to-the-internet-of-things-iot-are-you-ready.

24. Taber, Jennifer M, et al., "Why Do People Avoid Medical Care? A Qualitative Study Using National Data." *Journal of General Internal Medicine*, Springer US, Mar. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4351276/.

25. Team, DataFlair. "Artificial Neural Networks for Machine Learning - Every Aspect You Need to Know About." *DataFlair*, 6 Aug. 2019, data-flair.training/blogs/artificial-neural-networks-for-machine-learning/.

26. Team, DataFlair. "Artificial Neural Networks for Machine Learning - Every Aspect You Need to Know About." *DataFlair*, 6 Aug. 2019, data-flair.training/blogs/artificial-neural-networks-for-machine-learning/.

27. Teeuw, Micheal. "What the Best Slim LCD 24 24'+ Magic Mirror? I've Been Only Been Able to Get Components down to 1.5'." *MagicMirror Forum,* 15 Sept. 2017, forum.magicmirror.builders/topic/4847/what-the-best-slim-LCD-for-24-magic-mirror-i-ve-been-only-been-able-to-get-components-down-to-1-5.
28. Tugarmaes. "Get Help for Those Baggy Bags Under Your Eyes: Under Eye Wrinkles, Puffy Eyes, Under Eye Bags." *Pinterest,* www.pinterest.com/pin/647533252642877694/.
29. Verkruysse, W., Svaasand, L. O., & Nelson, J. S. (2008). Remote plethysmographic imaging using ambient light. Optics Express, 16(26), 21434. doi: 10.1364/oe.16.021434
30. Voss, Matthew. "Magic Mirror ." *GIthub.io,* 2017, mpvoss.github.io/mirror.html.
31. Yang, N., Zhao, X., & Zhang, H. (2012). A non-contact health monitoring model based on the Internet of things. 2012 8th International Conference on Natural Computation. doi: 10.1109/icnc.2012.6234771
32. Zhao, Tingting. *A Computer Vision Application for Assessing Facial Acne Severity from Selfie Images.* arxiv.org/ftp/arxiv/papers/1907/1907.07901.pdf.