Annotated Bibliography on AI, fairness, architecture

last edited: March 22, 2021

Abstract

We would like to develop classification of action potentials.

1 Introduction

It is a fact that racism exists in the health industry and affects billions of people around the world. It manifests as a systematic discrimination in countless facets and aspects of the health industry to the point of being supported through institutional policies. This means that it is not practiced through the aberrant behavior of a handful of individuals, although this remains possible, but through unintentional and unconscious bias based on negative stereotypes, and is supported institutional policies.

One noticeable example of this is the significant difference in infant mortality rates and pregnancy related deaths between various races. Between 1999-2013, infant mortality rates were highest among infants born to non-Hispanic black women with 11.11 infant death for ever 1,000 live births. Infants born to non-Hispanic Asian or Pacific Islander mothers experienced the lowest rates of 3.90. In 2014, non-Hispanic black mothers also had the highest percentage of preterm births of the five racial and ethnic groups evaluated in said study (non-Hispanic blacks, non-Hispanic American Indians or Alaska natives, Hispanic or Latina, non-Hispanic whites, and non-Hispanic Asians or pacific islanders) Black women were also found to be three to four times more likely to die in pregnancy related births when compared to white women.

Another example lies in the distribution of primary care physicians by zip code. Gaskins *et al.* found that the odds of a given zip code having a primary care physician shortage were 67 percent higher for majority African American zip codes but 27 percent lower for majority Hispanic zip codes. This number varied with the amount of segregation in said area: as segregation increased, so did the odds of having a shortage of primary care physicians. The inverse was true for primarily Hispanic and Asian zip codes.

One more example lies in the astonishing rate that minority medical patients are disbelieved or not taken seriously about their health. When polled about perceived pain in white and black patients, a study found that a substantial number of white laypeople and medical students and residents held false beliefs about biological differences between black and white patients. This phenomenon is particularly dangerous because it can cloud a health care provider's judgment and effect their treatment recommendations and diagnosis accuracy. In addition, Asian patients were found to be 1.21 times more likely than white patients to be admitted to the hospital following an ED visit and Black patients were 7% less likely to receive an urgent ESI score than white patients to immediate or emergent ESI scores, as opposed to semi- or non-urgent scores. Black patients were additionally 10% less likely than white patients to be admitted to a hospital and were 1.26 times more likely than white patients to die in the ED or hospital. While many of these studies were based in the United States, it is evident that this problem is international because Public Health England found that People of Chinese, Indian, Pakistani, Other Asian, Caribbean and Other Black ethnicity had between 10 and 50% higher risk of death from the COVID-19 when compared to White British, which mirrors a similar theme in the United States.

In addition to the statistics and examples listed here, one should wonder what additional ways has racism affected people that have been covered up, or simply not studied yet.

On a separate note, melanoma is a type of skin cancer that presents itself initially as a marking or deformity on the skin. In the year 2020, invasive melanoma is estimated to account for 1% of all skin cancer cases as well as the vast majority of skin cancer deaths. It is also estimated to be one of the most common cancer diagnoses for Americans aged 20-29 years old.

Racism can apply to the already deadly melanoma cancer in a number of ways Despite the occurrence of melanoma in non-Hispanic Black people, survival rates for them lag behind that if non-Hispanic White populations. Skin cancers in people of color often present atypically or in advanced stages when compared to Caucasian patients at the time of diagnosis, with post-metastization diagnoses at rates of 15% of Hispanics, 13% of Asians, and 12% of Black patients, compared with six percent of non-Hispanic Whites for men and seven percent of Hispanics, 21% of Asians, and 19% of Black patients were diagnosed with late-stage melanoma compared with four percent of non-Hispanic Whites. Out of patients who received surgical treatment for melanoma-related complications. The 10-year melanoma-specific survival was 73% lower in black patients than in white patients and other races who had survival rates of 88% and 85% respectively.

To solve the intersection of racism and its effects on melanoma patients, we propose using Artificial Intelligence algorithms (AI) and machine learning as a tool for detecting melanoma in its early stages. They have become increasingly more accessible due to recent advances in access to large data sets, fast computing, and cheap data storage, which has encouraged the development of machine learning algorithms with human-like intelligence in dermatology. However, to prevent hampering this tool's effectiveness and producing ill effects similar to that stated in previous findings, it is crucial that these algorithms are built to be free from racial bias, especially since melanoma presents on the skin. This can be done on training an AI based on melanoma cases with a variety of skin tones, equalizing these skin tones, and taking into account the relative sparsity of melanoma cases among minority patients.

2 Methodology

2.1 Pre-Processing Algorithms

2.1.1 CLAHE

Contrast-limited adaptive histogram equalization (CLAHE) is the current state-of-the-art method used for histogram equalization that addresses a lot of the issues other algorithms struggle with. Normal histogram equalization involves distributing the contrast values of the entire image evenly across the entire contrast spectrum. While this works for a uniform condensed distribution of contrast of the image, when there exists varying distributions of contrast across different sections of the image, normal histogram equalization fails to sufficiently equalize all sections of the image. This is where the adaptive part of the CLAHE algorithm is important, where instead of balancing color contrast based on the entire image, CLAHE breaks the image up into tiles that are subsections of the image. These tiles are then used to perform histogram equalizations on subsections of the image. Due to separating the image into smaller regions however, adaptive histogram equalization (AHE) algorithms suffer from a new problem of being more prone to noise in the image. This is where the contrast-limited portion of the CLAHE method is useful. In order to limit the over-amplification of contrast due to noise, CLAHE defines a threshold value for which only the highest parts of the histogram are clipped off and then equalized. The CLAHE algorithm requires 3 input parameters to operate on. The first one being the input image. This is generally a black and white image, however the CLAHE method can also be applied to colored images as well. In the case of colored images, instead of equalizing each of the 3 red, green, and blue spectrums, it is usually recommended to convert the image to the HSV color space and only apply CLAHE to the luminance channel. The other two inputs that were described above are the clip limit value and the grid size for the tiles. The CLAHE algorithm will then run the operations outlined above on the imputed image and output an image with equalized contrast.

2.1.2 Histogram Equalization in MatLAB

There are a variety of methods to accomplish Histogram Equalization in Matlab. The first of which includes the MatLAB Specific command, Histeq. This algorithm is particularly useful because it's easy to use and it works with little to no modifications, allowing the algorithm to be rather short. The command that this algorithm is based on is standard with Matlab's Image Processing Toolbox, and even allows users to input a target histogram should the user want to limit the number of "bins" produced. Despite these benefits, there are a number of caveats to using this algorithm. It can only work with one color channel at a time, so this algorithm would need be run on the red, green, and blue color channels of an image, and then the resulting three histograms could be composited into one equalized image. Unfortunately, this algorithm is difficult to adjust should the user want to alter the method used. This can be solved by building a histogram equalization method from scratch.Dr. Agrawal's algorithm is especially useful because it doesn't use the Matlab specific histogram equalization command, a user can adjust the algorithm as much as is necessary to achieve the desired result. This, of course, will require more expertise in histogram design and equalization methods, but this is worthwhile for the benefits of this algorithm.

2.1.3 LB-CLAHE

LB-CLASH is a machine learning based version of CLASH, it seems to have slightly better results than the standard version of CLASH, but there is currently only one article about the topic (1). Machine learning was used in the artial to control and optimize the hyperparameters for CLASH histogram equalization. The Authors found that LB-CLASH was able to quickly suggest the hyper parameters for a CLASH based histogram equalization that provide better image results. Machine learning does not do the histogram equalization; it just suggests parameters. The article asserts that this method of selecting the hyper parameters is more effective than the other options. They do provide the source code for this project (2), along with the datasets that they used. This method may be usable for our purposes but it will increase the time to compute because before CLASH can be run it must first have its hyper parameters have been determined by LB-CLASH.

2.2 Classification Algorithms

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are used to classify data based on their internal knowledge base acquired from training on past data. CNNs as a whole is a large topic, so this writeup will mainly focus on the aspects that are relevant to image classification. Functionally, a CNN is a structure of nodes similar to Artificial Neural Networks (ANNs), so it is important to understand how ANNs work before diving into CNNs. ANNs have 3 layers of nodes, the first being the input layer where each predictor for the classification is inputted into the system. ANNs then have an output layer and some number of hidden layers containing perceptrons. Each of these perceptrons have an activation function where, depending on the input values and their corresponding weights, the perceptron outputs a corresponding classification. These perceptrons together as a network output an overall classification for the data from the output layer. The reason why training data is required for neural networks is because initially the weights for how much each predictor or input value should be considered is unknown. So by showing the model a large sample size of pre-classified training data allows it to determine the optimal weights for each predictor or input value in order to achieve the correct classifications. CNNs depend on this fundamental network of perceptrons, however they take it a step further with convolutional layers. While ANNs can technically be trained to perform image processing and classification problems, it would require an unwieldy amount of weights and layers as every pixel in the image would have to be an input. With convolutional layers however, the amount of required parameters is significantly reduced as the image is split up into tiles. The size of the tile is defined by the model's batch size. In summary, the notable hyperparameters for initializing a CNN model include: number of convolutional layers, initial weights, type of activation function, number of epochs (number of iterations the model trains on the data), and batch size. Once the model has been defined, it is then trained and tested on a pre-classified database until a sufficient success rate is achieved. Finally once the model is trained, the only step required to use it is to input a given image, the CNN model will process the image, and then the model will output a corresponding classification with a given degree of certainty.

2.2.2 Edge Detection

MatLAB offers a selection of edge detection algorithms, which may be useful in detecting the boundaries of an instance of Melanoma.t relies on discontinuities in brightness to detect an “edge”. There are several algorithms that can be used to detect these edges. The examples given use both the Sobel and Canny methods This algorithm has demonstrated promising results on very clearly defined melanoma occurrences. Further testing of this algorithm is required for further confirmation on this. However, this algorithm has demonstrated weaknesses when used on images with textured skin. Applying a blurring filter to an image can improve on this weakness, but a better system to remove skin texture is necessary for this algorithm to improve. Further investigation into which edge detection algorithms work best for certain instances of melanoma certainly deserve further investigation.

2.2.3 Haar Cascade

Haar Cascade is a machine learning object detection algorithm conceptualized by Paul Viola and Michael Jones in 2001 The algorithm is used to identify objects, specifically faces and body parts, however it can be implemented to identify many other objects, including melanoma in skin cancer patients. There are four stages to the algorithm: Haar Feature Selection, Creating Integral Images, Adaboost Training, and Cascading Classifiers. Step one in the algorithm is to collect the haar features, which are adjacent rectangular regions at a specific location in a detection window, using Integral Images makes feature collection fast and easy. The next step is to implement the Adaptive Boosting meta algorithm to make sure that the classifiers select the best features available. Finally, in the Cascading Classifiers, each stage consists of decision dumps that decide whether or not there is an object within the window that they're analysing. If there is an object then the detector will mark it as a positive detection, if there is no object then the detector will mark it as negative. Sometimes a detector will incorrectly detect a positive or negative object. In order to combat this multiple stages are implemented to properly filter out false positives and negatives. To test the algorithm the user must give the program a large quantity of positive and negative objects, which would be images of patients with and without melanoma in order to "train" the algorithm to know what to look for when analyzing images. This algorithm may have a hard time properly detecting melanoma in skin cancer patients because it can look very different from patient to patient, however, if enough images are given this algorithm could be very promising in its ability to "learn" how to detect a given object. If it is given enough images of positive and negative images, like images of moles and malignant tumors and other growths, this algorithm would have no trouble properly detecting melanoma in skin cancer patients.

2.2.4 Histogram of Oriented Gradients

Histogram of Oriented Gradients(HOG) is an algorithm founded by Robert K McConnell in 1986 but was popularized by French researchers Navneet Dalal and Bill Triggs. The algorithm is designed to take an image, filter it, and categorize it by its distribution of directions of gradients. In order to do this, HOG first reshapes all the images to be the same dimensions and normalizes them to get rid of any possible illumination effects. Next the gradient images are calculated by finding the magnitude and direction of each gradient, using g=gx2+gy2and=arctan(gy gx), respectively. The next step is to calculate the histogram of the gradients using 8x8 cells. Afterwards is the 16 x 16 block normalization which is done by concatenating 4 8x8 cells into a 36 x 1 vector and dividing each element in the vector by its length. The length of a vector is calculated by squaring each element in the vector and adding them together and then finding the square root of that sum. Finally, the HOG feature vector is calculated by concatenating 105 positions of 36x1 vectors together to get a 3780 dimensional vector. In order to run the algorithm, an image must be given. Based on the images given the code will resize the image to be 32x32 if it isn't already, puts it in greyscale, then a histogram is calculated and a 4x4 matrix is designed for each cell. This algorithm could be useful towards accurately pointing out what patient has melanoma and what patient does not. Since melanoma has distinguished features they could be picked up by the algorithm and accurately categorized in a bin. This algorithm can also be “trained” on finding specific features so if enough images are put into the algorithm they could be accurately categorized based on if they meet the criteria for melanoma.

2.2.5 Support Vector Machine

Support Vector Machine (SVM) Is a form of supervised machine learning that is used for classification. Basically, SVM tries to find a boundary/ space between two types of data. In SVM, we plot the data in the data set in a N dimensional space where N is the number of attributes in the data set. Then, we try to find the hyperplane/boundary that separates the data. This boundary can be linear and can consist of a regression line, or the boundary can be nonlinear and can divided data into kernels (like k-means clustering). SVM is used to classify a data set. For example, one part of the data can be classified as ‘yes’ the other can be known for a question. According a study done on cancer patients undergoing mastectomies, they utilized machine learning to predict post-operative outcomes based on patient reports and SVM and neural networks were the most accurate. However, SVM had smaller confidence interval than neural networks.

2.3 Enviroments

In order to support operation of our algorithms, we investigated and setup several programming environments. These included Google Cloud, Amazon Web Services, and a physical machine on campus.

3 Status of References

Reference	Status