***Emotion Detection of Autism Patients***

*Objective*

The objective of this project is to create a machine learning model to detect emotions of people diagnosed with Autism from the expressions of their face. We are trying to detect mainly six emotions: Happy, Sad, Disgust, Fear, Angry, Surprise and Neutral.

*Abstract*

Autism spectrum disorder refers to a neurodevelopmental disorder characterized by repetitive behavior patterns, impaired social interaction, and impaired verbal and nonverbal communications. Autism spectrum disorder begins in early childhood and eventually causes problems functioning in society — socially, in school and at work. A child or adult with autism spectrum disorder may have problems with social interaction and communication skills they lack the ability to convey what their likes and dislikes are to other people and also the care takers or the family of people who are diagnosed with such a disease also have a hard time understanding their emotions. Building a machine learning model to recognize the emotion of a person diagnosed with autism spectrum disorder will be helpful for that person as well as the family of caretaker of the diagnosed person.

*Features*

Upon Completion of the project, the model will be able to predict the emotion of a person.

Model will be able to detect if a person is happy, sad, Disgust, fear, angry, surprise or neutral.

Input to the model can be an image, video or live video. This project can be used in hospitals, orphanage or by people who are in regular contact with a person diagnosed with autism, so they can understand the emotion of a person diagnosed with autism.

*Technology Used*

* Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is open-source, so it is free to use, modify and distribute the python source code.

Python is a high-level programming language that has English-like syntax making it easier to read and understand the code also the standard library of python is very big, that any and all function needed for a project can be found minimizing the use of external libraries. Different python packages like Pandas, NumPy will used in the project.

* Machine Learning

Machine learning is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications where it is difficult to develop conventional algorithms to perform the needed tasks.

The model which predicts the emotions of people will be built using machine learning algorithm.

* OpenCV

OpenCV or Open-Source Computer Vision Library is a library of programming functions mainly aimed at real-time computer vision. OpenCV is a great tool for image processing and performing computer vision tasks. it is an open-source library that can be used to perform tasks like face detection, object tracking, landmark detection etc. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis of images or videos.

OpenCV consider image as a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be and uses this data to analyze the image, alter the image or extract features from the image.

*Libraries Used*

* NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high -level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

* Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

*Algorithm Used*

The Haar-Cascade method is used for this project. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. We will use Haar-Cascade method to detect faces from a given input. The algorithm will detect the faces in every frame of the webcam feed and the region of the image containing the face is then resized to 48x48 and is then passed as input to

the CNN. The network outputs a list of softmax scores for seven different classes of emotions and the emotion with the highest score is then displayed on the screen.

The first step is to collect the Haar features. A **Haar feature** is essentially calculations that are performed on adjacent rectangular regions at a specific location in a detection window. The calculation involves summing the pixel intensities in each region and calculating the differences between the sums. These features can be difficult to determine for a large image.

This is where the second step **integral images** come into play. The number of operations is reduced using the integral image.  Integral images essentially speed up the calculation of these Haar features. Instead of computing at every pixel, it instead creates sub-rectangles and creates array references for each of those sub-rectangles. It’s important to note that nearly all of the Haar features will be **irrelevant** when doing object detection, because the only features that are important are those of the object.

The third step pf Haar-Cascade method is **Adaboost**. We use Adaboost to obtain the relevant features of an object. Adaboost essentially chooses the best features and trains the classifiers to use them. It uses a combination of **“weak classifiers”** to create a **“strong classifier”** that the algorithm can use to detect objects. Weak learners are created by moving a window over the input image, and computing Haar features for each subsection of the image. This difference is compared to a learned threshold that separates non-objects from objects. Because these are “weak classifiers,” a large number of Haar features is needed for accuracy to form a strong classifier.

The last step combines these weak learners into a strong learner using **cascading classifiers**. The cascade classifier is made up of a series of stages, where each stage is a collection of weak learners. Weak learners are trained using boosting, which allows for a highly accurate classifier from the mean prediction of all weak learners.

Based on this prediction, the classifier either decides to indicate an object was found (positive) or move on to the next region (negative). Stages are designed to reject negative samples as fast as possible, because a majority of the windows do not contain anything of interest. It’s important to maximize a **low false negative rate**, because classifying an object as a non-object will severely impair your object detection algorithm

*Data Collection*

Data is mainly collected from internet. Relevant data can be obtained from websites like Kaggle, GitHub etc.