**BrainDead**

**“Report on Emotion Detection”**

**Team (The avengers)**

Members (1. Nikhil S ,2. Pritam M)

**Department of Computer Science and Technology, IIEST Shibpur.**

**Declaration**

I Nikhil S and Pritam M students of 2nd year B Tech Computer Science Engineering from CMR University, Bangalore hereby declare that the project work entitled “Emotion Dtection” has been carried out by us. This report is submitted in partial fulfillment of the requirement for the event. We declare that the report submitted is to the best of our knowledge.

**Index**

|  |  |  |
| --- | --- | --- |
| Sl No | Topics | Page NO |
| 1. | Introduction | 4 |
| 2. | Libraries | 5 |
| 3. | Results and conclusion | 8 |
| 4. | Annexure | 9 |

**Introduction**

So looking into the present times, detecting whether a person is happy, sad or neutral is very difficult. You never know what a person is going through just by looking into his face. We need to talk, sometimes even by talking you never know. So we tried to create a program in which emotion of a person can be judged just by seeing a picture of him. Mostly emotions are of three types, either happy, sad or neutral.

We know that no machine can judge these things, but we tired to some extent.

We believe we may be correct sometimes in few cases.

**Libraries used**

**1.OS**

The "os" library in ML usually refers to the "os" module in Python, which provides a way to interact with the operating system. It allows you to perform various tasks related to file handling, directory operations, and process management. Here are some of the commonly used functions in the "os" library:

1. os.listdir(): This function returns a list of all files and directories in the specified path.
2. os.getcwd(): This function returns the current working directory.
3. os.chdir(path): This function changes the current working directory to the specified path.
4. os.path.join(): This function joins two or more path components and returns a single path.
5. os.path.exists(path): This function checks if the specified path exists or not.
6. os.mkdir(path): This function creates a new directory at the specified path.
7. os.rmdir(path): This function removes the specified directory.

The "os" library can be useful in ML for various tasks, such as loading data from files, creating new directories for saving model checkpoints, and managing the environment variables required for running ML applications.

**2.Glob**

The "glob" library in ML refers to the "glob" module in Python, which provides a way to search for files and directories based on specified patterns. The module uses Unix shell-style wildcards to match the file or directory names.

Here are some commonly used functions in the "glob" library:

1. glob.glob(pattern): This function returns a list of file or directory paths that match the specified pattern. The pattern can include wildcard characters such as \* (matches any string) and ? (matches any single character).
2. glob.glob(pathname, recursive=True): This function recursively searches for files and directories that match the specified pattern in the given pathname.
3. glob.has\_magic(pattern): This function checks whether the specified pattern contains any wildcard characters.

The "glob" library is often used in ML for loading data from multiple files that match a certain pattern, such as loading multiple images or text files for training a machine learning model. It can also be used to search for log files or model checkpoints based on a specific naming convention.

**3.TensorFlow**

The TensorFlow library is an open-source machine learning framework developed by Google. It is used for building and training various types of machine learning models, including neural networks, deep learning models, and reinforcement learning models.

Some of the key features of the TensorFlow library include:

1. High-level APIs: TensorFlow provides high-level APIs, such as Keras and Estimators, that make it easier to build and train machine learning models.
2. Flexibility: TensorFlow supports various deployment options, including running models on CPUs, GPUs, and TPUs, as well as on mobile and embedded devices.
3. Distributed training: TensorFlow supports distributed training, allowing you to train large models on multiple devices or clusters.
4. Visualization tools: TensorFlow provides tools for visualizing and debugging machine learning models, including TensorBoard, which helps you visualize the model graph, monitor training progress, and analyze performance.
5. Community support: TensorFlow has a large and active community of developers who contribute to the library, provide support, and share resources and best practices

**4.Keras**

The Keras library is an open-source neural network API written in Python. It is designed to be a user-friendly and modular deep learning library that can run on top of other popular machine learning frameworks such as TensorFlow, Theano, and Microsoft Cognitive Toolkit. Keras provides a high-level API for building and training neural networks, making it easier to implement complex models with fewer lines of code.

Here are some key features of the Keras library:

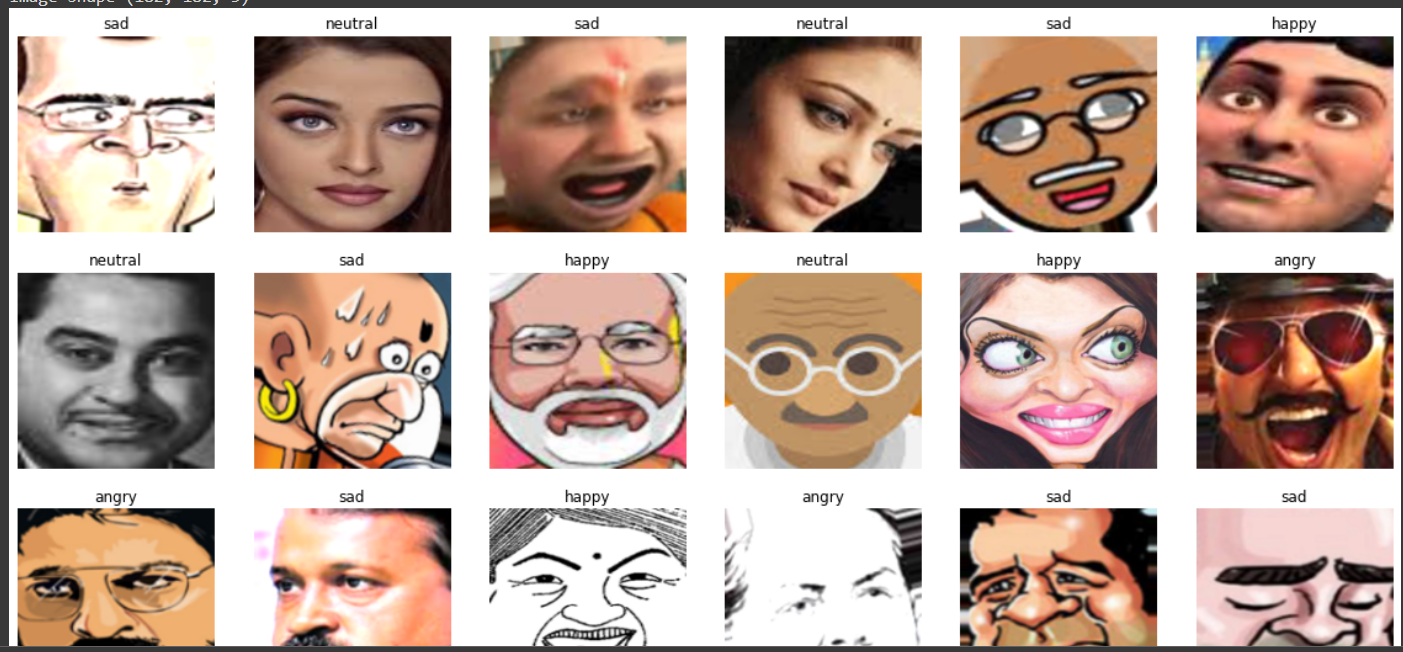
1. User-friendly API: Keras provides a simple and intuitive API for building and training neural networks, allowing users to quickly prototype and experiment with different models.
2. Modular architecture: Keras is built with a modular architecture, which allows users to easily combine different layers and modules to create custom models.
3. Extensibility: Keras can be extended with custom layers, loss functions, and other components, making it a flexible and powerful library for deep learning.
4. Backward compatibility: Keras provides backward compatibility with previous versions, making it easier to upgrade to newer versions without breaking existing code.
5. Cross-platform support: Keras can run on a variety of platforms, including CPUs, GPUs, and TPUs, and can be used for both research and production environments.

Keras is widely used in ML for a variety of applications, such as image recognition, natural language processing, and speech recognition. It is also used by researchers and developers for prototyping and testing new neural network architectures

**Results and conclusion**

So, we had tried a few times and achieved the desired results.

Just for example, if we upload an picture of a person our program would say whether he/she is happy, sad or neutral. The result may be shown in form of percentage. Once we get the result we can know the emotion of a person.



**References**

1. <https://www.google.com/>
2. <https://www.youtube.com/>
3. https://citrusbug.com/blog/python-libraries-for-machine-learning

**Annexure**

<https://colab.research.google.com/drive/1b2Ykczxsxt1OLWLZ3sHhd4SrZxi2DvgU?usp=sharing>