**ABSTACT**

Based on a well-known benchmark, a comparison between the present study and the literature was carried out. This project investigates a variety of Machine Learning (ML) and Deep Learning (DL) algorithms for classifying emotional events using EEG brainwave data. We propose in this project an emotional recognition system based on physiological signals. We adopt the three basic emotions that are: neutrality, positive and negative. An experiment has been conducted to verify the feasibility of the proposed system. This experience has allowed us to acquire EEG signals and to create an emotional datasets. Thereafter, we have chosen the CNN techniques to classify the EEG images and to analyze the results.

Recognizing emotions using EEG (Electroencephalography) signals is a fascinating area of research at the intersection of neuroscience, psychology, and computer science. EEG measures the electrical activity of the brain through electrodes placed on the scalp, providing a real-time glimpse into brain function.

Emotions play a crucial role in human behavior and communication. Understanding and interpreting emotions accurately can significantly impact various fields, including healthcare, human-computer interaction, and psychology. Traditional methods of emotion recognition rely on self-reporting, facial expressions, or physiological signals like heart rate and skin conductance. However, EEG offers a unique window into the brain's activity, providing insights that complement other modalities

Emotions elicit distinct patterns of brain activity, making EEG a promising tool for emotion recognition. Studies have identified specific EEG features associated with different emotional states, such as happiness, sadness, fear, and anger.

**INTRODUCTION**

In our daily activities such as thought function, conversation, and decision-making, emotions play a significant part. It’s also really important for human-machine interaction. Recognition of emotions has gained considerable interest from various study groups, documenting the mental condition of the individual, such as facial gestures, speech, and body language, both of which are new forms of identifying human emotions. Recent years have seen a growing focus being given to physiological signs dependent on emotion identification. Most of the approaches focused on physiologic signals utilize well-designed classifiers with hand-crafted features to identify human emotions.

Electroencephalograph (EEG), the measurement of electrical stimulation in the brain, is a commonly used measure of brain function during cognitive activities such as working memory (Bird et al , Bird et al., Nie et al, Jatupaiboon et al , Shah et al, Bird et al . Identify invariant representations to inter-and intrasubject variations and also intrinsic noise associated with these data represent challenges for EEG data modeling for cognitive activities. Plutchik et al, proposed eight different states of emotion such as anticipation, disgust, sadness, fear, anger, and joy.

Ekman, proposed six specific states of emotion based on facial expressions like happiness, surprise, disgust, fear, sadness, and anger. All belong to the distinct scheme of classification of emotions. And the most frequently used paradigm of classification of valence and intensity feelings, introduced by Russell, belongs to the aspect method. Electroencephalography (EEG) is a brain exploration method that uses electrodes mounted on the scalp for brain electrical activity monitoring, often shown in the form of a line called an electroencephalogram. Brain wave signals can be recorded by these electrodes and each one of them has its proper characteristics as band powers and mental state descriptions. For instance, brain waves can be Delta waves that are between 0.5Hz and 4Hz and they refer to deep sleep, Theta waves that are between 4Hz and 7Hz and they refer to light deep meditative state and hypnosis, Alpha waves that are between 7Hz and 13Hz and they refer to relaxed alertness and light meditative state, Beta waves that are between 13Hz and 30Hz and they refer to active and vigilant state and finally, Gamma waves that are between 30Hz and 100Hz and they refer to intense neuronal activity and hyper vigilance. While the deep neural networks have successfully implemented multiple images, video, and text data on a wide scale, in the field of neuroimaging they remain largely unexplored. Indeed, one of the key causes for this is that the volume of samples in most datasets on neuroimaging is limited, rendering these data less suitable for training large- networks with millions of parameters. Let us take the example of a BCI based on 978-1-6654-1224-7/21/$31.00 ©2021 IEEE Motor Imagery (MI) dependent on oscillatory activity, i.e. a BCI that can identify imagined movements such as left-hand or right-hand imagined movements.

Electroencephalography (EEG) offers a non-invasive method to measure brain activity through electrodes placed on the scalp. It provides real-time insights into brain function by recording electrical signals generated by neuronal activity. Emotions evoke distinct patterns in EEG signals, making it a promising avenue for emotion recognition. Despite challenges such as noise and individual differences, advances in signal processing and machine learning are improving accuracy. EEG-based emotion recognition has applications in healthcare, human-computer interaction, and beyond, offering insights into human behavior and facilitating innovative technologies.

**LITERATURE SURVEY**

In the literature, many researchers have worked on emotion recognition based on EEG signals. Some of these studies use artificial intelligence (AI) to improve emotional behavior classification performance utilizing Machine Learning (ML) and Deep Learning (DL) methods. In this section, we present some related works. Bird et al, [1] evaluated the results of various combinations of selection algorithms and classifier models in terms of recognition accuracy and the number of features predicted from the five EEG headband signals: alpha, beta, theta, delta, and gamma. They researched to develop discriminative EEG-related features and correct classification methods that could categorize brainwave signals depending on their degree of activation or frequency for useful for human-machine interaction to identify the mental state. They used classic classifiers such as Bayesian Networks, Help Vector Machines, and Random Forests to obtain an average accuracy of more than 87%.

To do this analysis, we have conducted in-depth research on two papers; Bird et al, and Bird et al, [6] from which we use their benchmark to add some hangings in the preprocessing and processing processes. To estimate the network’s optimal hyper parameters before classification, Bird et al, [2] used an evolutionary approach to evaluate Multilayer Perceptron (MLP). In their paper, they explored DL and tuning with Long Short-Term Memory (LSTM), and measures Adaptive Boosting of the two types of models for each issue. Three tests are presented for comparison using different classifiers: the first for attention state classification (https://www.kaggle.com/birdy654/ eeg-brainwave-dataset-mental-state), the second for emotional sentiment classification (https://www.kaggle.com/birdy654/ eeg-brainwave-dataset-feeling-motions), and the third experiment to estimate the amount a subject think about (http:// www.mindbigdata.com/opendb/). They used the Muse headset to gather data from the TP9, AF7, AF8, and TP10 extracranial electrodes, as well as the MindBigData dataset to collect data from the TP9, FP1, FP2, and TP10 electrode locations. They used for feature extraction and selection, temporal statistical features by using The Logarithmic Covariance matrix model to a log-cov vector and thus statistical features and Fast Fourier Transform (FFT) for spectrum analysis.

A bioinspired evolutionary algorithm was used to pick the attributes. They then employed a Multilayer Perceptron (MLP) with up to 100 neurons per layer and a maximum of three hidden layers for processing. It was trained for 50 epochs to predict the class for each number of units on a layer, and manually optimized topology was created by feeding a vector of time sequence statistical data in batches of 50 data points. Finally, an AdaBoost algorithm was used for both algorithms to mitigate both the ill effects of manually optimizing the LSTM topology as well as fine-tune the models overall. First, the use of an AdaBoosted LSTM, which, while it took more time and resources to train compared to other methods, managed to achieve 84.44% and 97.06% accuracy for the first two datasets (mental and emotional state classification).

Second, an AdaBoosted Multilayer Perceptron was optimized using an evolutionary hyperheuristic algorithm. It took less time to practice, although its classification accuracy was marginally lower than that of the AdaBoosted LSTM (79.7% and 96.23% for the same two experiments). LSTM can achieve 84.44%, 97.06% accuracy, and 9.94% accuracy, respectively, on the attention, emotion, and number datasets. For the first two experiments, they used an evolutionary-optimized MLP and the Adaptive Boosted LSTM and have close results but significantly higher for the evolutionary-optimized MLP for the number-guessing experiment with an Adaptive Boosted DEvo MLP reaching 31.35%, while being significantly faster to train and classify. In particular, the unboosted DEvo MLP’s accuracy in the same benchmarks was 79.81%, 96.11%, and 27.07%.

Bird et al, explored the application of single and ensemble classification methods to take windowed data from four points on the scalp and quantify the data into an emotional representation of what the individual was thinking at the time. Methods show that using a low resolution, commercially available EEG headband can be effective in classifying the patient’s emotional state. They worked on a dataset for emotion classification (https://www.kaggle.com/birdy654/ eeg-brainwave-dataset-feeling-emotions) based on emotional states (Positive, Neutral, and Negative). Six minutes for each state were recorded from two adults, 1 male and 1 female aged 21 ± 1 producing a total of 36 minutes of brainwave activity data for each of the 6 film clips (3 positives and 3 negatives). To avoid contamination by the latter, they gathered neutral data without stimulation and before any of the emotions data for a third class, which would be the subject’s resting emotional state. Three minutes of data were collected each day to avoid the interference of a resting emotional state. Three minutes of data were gathered each day to reduce the interference of a resting emotional state. The TP9, AF7, AF8, and TP10 extracranial electrodes are monitored using the Muse headset. They worked on single model algorithms: OneR (One Rule), Random Tree (RT), Sequential Minimal Optimisation (SMO), Naive Bayes (NB), Belief Network (BN), Linear Regression (LR), Multilayer Perceptron (MLP), and ensemble model algorithms: Random Forest (RF), Vote, Adaptative boosted RF (AB(RF)). For features selection, InfoGain performed better as an evaluator than OneR, BayesNet, and Symmetrical uncertainty. The best model was applying Random Forest on relevant attributes from the Infogain evaluator with an accuracy 2 of 97.89%. The multilayer perceptron was consistently the best model for single classification, demonstrating the usefulness of neural networks for this problem. InfoGain is the measurement of changes in entropy after segmenting a dataset based on an attribute.

**PROBLEM STATEMENT**

Emotion classification is a major task, which needs to be analysis psychological signals of the human. And this analysis cannot be identified by everyone. This is possible only by psychologists who understand the human behaviour. If someone is suffering from mental disorders, or by suffering from body stroke or any other mental issues due to accidental cases, their human won’t be able to share their feelings, but their nervous system can react for situations which is happening in front of them. In all these situations a common human cannot understand. And this problem can be solved by these kinds of application which tells whether patient’s nervous system is in working condition, which may help to give better treatments and get back to the normal stage.

**Objectives**

**The main objective of this study is:**

(1) Design and validate a new framework for automatic classifying emotions on the bases of images as accurately as possible.

(2) Explore knowledge of the significance of the application of the feature extraction method on applying a deep learning algorithm.

(3) Investigate a sustainable classification model for the proposed features to differentiate the subject groups.

(4) Improve classification accuracy compared to existing methods as the deep learning method automatically optimizes the parameters and requires less prior expert knowledge for the feature extraction procedure to perform effectively

(5) Build a low cost time model. To the best of our knowledge, this is the first work to apply the deep leaning technique with a feature extraction method and without a feature extraction method for classification of emotions subjects from EEG data and images.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

In Existing system used 30 principal components from the result of PCA method and then 64 channels from raw EEG dataset in two separate experiments. The number of neurons in the output layer depends on the number of desired classes. In this study we need to determine if a person's behaviour if he is mind. Intermediate or hidden layers are useful to increase the ability of the network; and MLP can have multiple intermediate layers and there are no rules on the number of layers and nodes needed. Large numbers of hidden layers and neurons increase the complexity of the network and execution time and small numbers of layers and nodes lead to errors and low performance and poor generalization.

**Introduction to Existing EEG Signal Methods:**

Electroencephalography (EEG) signals have been extensively studied and analyzed. Using various methods to extract meaningful information about brain activity. These methods encompass signal processing techniques, machine learning algorithms, and statistical analyses, each serving distinct purposes in understanding and interpreting EEG data.

**Signal Preprocessing:**

Before analyzing EEG signals, preprocessing steps are often applied to enhance signal quality and remove artifacts. These include filtering to remove noise, artifact removal techniques such as Independent Component Analysis (ICA) to separate EEG components from artifacts like eye blinks or muscle activity, and segmentation to divide continuous EEG recordings into smaller segments for analysis.

**Feature Extraction:**

Feature extraction involves identifying relevant characteristics or patterns within EEG signals that are indicative of different brain states or cognitive processes. Common features include spectral power in different frequency bands (delta, theta, alpha, beta, gamma), connectivity measures such as coherence or phase synchronization between EEG channels, and event-related potentials (ERPs) in response to specific stimuli.

**Machine Learning Algorithms:**

Machine learning algorithms play a crucial role in EEG analysis by learning patterns from labeled data and making predictions or classifications based on unseen data. Supervised learning algorithms, such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are commonly used for tasks like emotion recognition, cognitive state classification, and brain-computer interface (BCI) applications.

**Statistical Analysis:**

Statistical methods are employed to identify significant differences or relationships in EEG data. These include traditional statistical tests such as t-tests and ANOVA for group comparisons, as well as more advanced techniques like permutation tests, cluster-based analysis, and machine learning-based feature selection methods to identify discriminative features for classification tasks.

**Disadvantages of existing system**

* As there are no rules to determine the best topology, therefore, it is only found by trial and error
* Existing methods like PCA and SVM are not accurate and time taken for processing and prediction is high and accuracy is less.
* EEG measures electrical activity at the scalp surface, which limits spatial resolution. It cannot pinpoint the exact location of neural activity within the brain, making it challenging to localize specific emotional responses to precise brain regions.
* EEG signals are susceptible to various types of noise and artifacts, including muscle movements, eye blinks, and environmental interference. These artifacts can distort the underlying neural signals, reducing the accuracy of emotion recognition algorithms.
* Emotions are complex and multidimensional, making it difficult to categorize them into discrete categories based solely on EEG signals. EEG may capture broad patterns associated with emotional arousal but might struggle to differentiate between subtle variations within emotional states.
* Analyzing EEG data requires expertise in signal processing, machine learning, and neuroscience. Interpreting EEG features in the context of emotions requires careful consideration of various factors, including experimental design, task complexity, and individual differences.
* EEG-based emotion recognition raises ethical concerns related to privacy and data security. Collecting and analyzing neural data to infer emotions could potentially infringe upon individuals' privacy rights and raise questions about consent and data ownership.

**PROPOSED SYSTEM**

In this project we are using CNN model for prediction of classification using machine learning we are using EEG emotional images dataset which is collected trained data using CNN model and model is saved. For user friendly detection web application using flask is designed where user can upload signal image using flask framework and check classification type as negative, positive and neutral also if negative it notifies through email to concerned authorities or doctors.

**1. Multi-Modal Data Acquisition:**

Integrate EEG signals with other physiological signals such as heart rate variability (HRV), facial expressions, and subjective self-reports to capture a comprehensive picture of emotional states. Combining multiple modalities can enhance the accuracy and robustness of emotion recognition algorithms.

**2. Advanced Signal Processing:**

Employ sophisticated signal processing techniques to enhance EEG data quality and extract informative features. This includes artifact removal methods, adaptive filtering, and time-frequency analysis to capture dynamic changes in brain activity associated with different emotional states.

**3. Feature Selection and Fusion:**

Identify discriminative features from EEG signals and other modalities using advanced feature selection algorithms. Explore feature fusion techniques to combine information from multiple sources effectively, leveraging the complementary nature of different modalities for improved emotion recognition performance.

**4. Machine Learning Models:**

Utilize machine learning algorithms to build emotion recognition models capable of learning complex patterns from multi-modal data. Explore deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture temporal dependencies and hierarchical representations in EEG signals.

**5. Personalized Emotion Models:**

Develop personalized emotion models that account for individual differences in brain activity and emotional responses. Incorporate subject-specific calibration and adaptation techniques to tailor the emotion recognition system to each user's unique physiological and cognitive characteristics.

**6. Real-Time Processing and Feedback:**

Implement real-time processing capabilities to provide immediate feedback on users' emotional states. This could involve developing user-friendly interfaces or integrating the emotion recognition system with interactive applications such as virtual reality environments or affective computing systems.

**7. Ethical Considerations:**

Address ethical considerations related to data privacy, informed consent, and responsible use of neurotechnologies. Ensure transparency and accountability in data collection, processing, and interpretation, and prioritize user autonomy and privacy rights throughout the development and deployment of the emotion recognition system.

**8. Validation and Evaluation:**

Conduct rigorous validation and evaluation studies to assess the performance and reliability of the proposed emotion recognition system. Validate the system across diverse populations and contexts to ensure its generalizability and effectiveness in real-world settings.

By integrating these components into a comprehensive framework, the proposed system aims to overcome the limitations of existing approaches and advance the field of emotion recognition using EEG signals, facilitating applications in healthcare, human-computer interaction, and beyond.

**Advantages**

* The proposed system was developed taking in mind the benefits of the medical sector to treat the patients mental illness.
* The developed system can detect classification in humans emotions and also provide the best output to identify their mind states.
* By proper knowledge of the patience health and the remedy can be taken for improving the health of the humans.

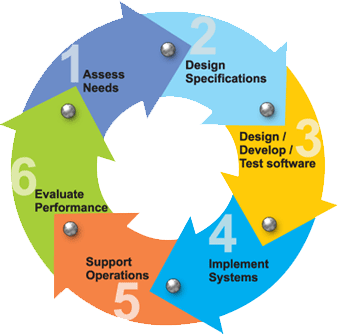
**DisAdvantages**

* **I**ntegrating multiple modalities, advanced signal processing techniques, and machine learning models can increase the complexity of the system and require significant computational resources.
* Complex machine learning models, such as deep neural networks, may lack interpretability, making it challenging to understand how the system arrives at its predictions. Lack of transparency in the decision-making process could raise concerns about trust, accountability, and user autonomy, particularly in sensitive applications like healthcare.

**Environment**

**Software Development Life Cycle**

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development.



**Requirements:**

Business requirements are gathered in this phase.  This phase is the main focus of the project managers and stake holders.  Meetings with managers, stake holders and users are held in order to determine the requirements.  Who is going to use the system?  How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase.  This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work.  The overall result is the system as a whole and how it performs, not how it is actually going to do it.

**Design:**

The software system design is produced from the results of the requirements phase.  Architects have the ball in their court during this phase and this is the phase in which their focus lies.  This is where the details on how the system will work is produced.  Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

**Implementation:**

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle.  For a developer, this is the main focus of the life cycle because this is where the code is produced.  Implementation my overlap with both the design and testing phases.  Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

**Testing:**

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.  Unit tests and system/acceptance tests are done during this phase.  Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model.  Now let’s delve into some of the traditional and widely used variations.

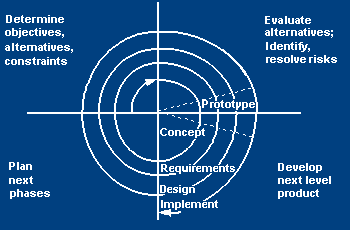
**SDLC METHDOLOGIES:**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

**The following diagram shows how a spiral model acts like:**



**The steps for Spiral Model can be generalized as follows:**

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning a designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested.Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

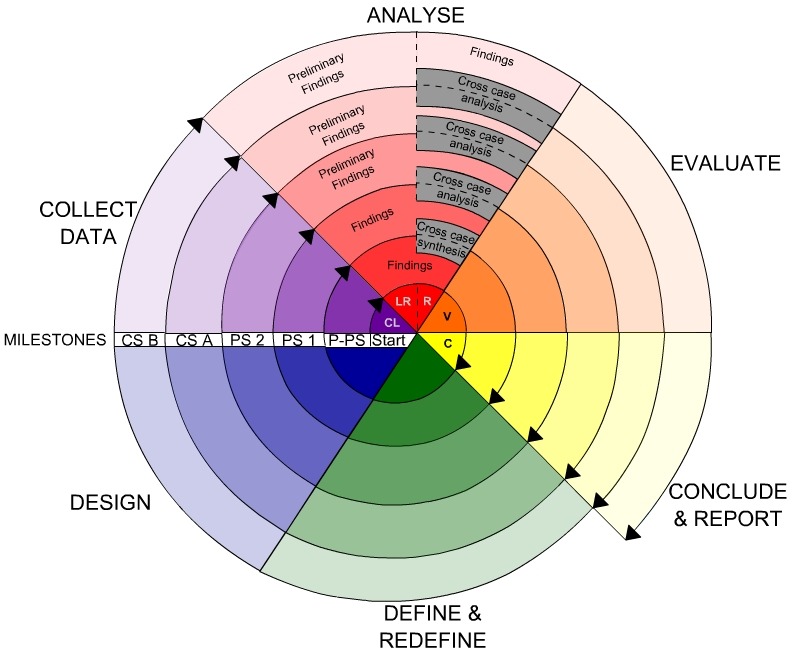
**STUDY OF THE SYSTEM**

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI’s at the top level has been categorized as follows

1. Administrative User Interface Design
2. The Operational and Generic User Interface Design

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.



**INPUT AND OUTPUT**

**INPUT DESIGN**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**INPUT STAGES:**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**INPUTMEDIA:**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.

**OUTPUT DEFINITION:**

# The outputs should be defined in terms of the following points:

* + - Type of the output
    - Content of the output
    - Format of the output
    - Location of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

**Functional requirements**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization,.
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.
* Understanding user’s preferences, expertise level and his business requirements through a friendly questionnaire.
* Input data can be in four different forms - Relational DB, text files, .xls and xml files. For testing and demo you can choose data from any domain. User-B can provide business data as input.

**Non-Functional Requirements:**

1. Secure access of confidential data (user’s details). SSL can be used.
2. 24 X 7 availability.
3. Better component design to get better performance at peak time

**FEASIBILITY STUDY:**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Economical Feasibility
* Operation Feasibility

**Technical Feasibility:**

In the feasibility study first step is that the organization or company has to decide that what technologies are suitable to develop by considering existing system.

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Here in this application used the technologies like Visual Studio 2012 and SqlServer 2014. These are free software that would be downloaded from web.

Visual Studio 2013 –it is tool or technology.

**ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

### **Determining** **Economic Feasibility:**

Assessing the economic feasibility of an implementation by performing a cost/benefit analysis, which as its name suggests compares the full/real costs of the application to its full/real financial benefits.  The alternatives should be evaluated on the basis of their contribution to net cash flow, the amount by which the benefits exceed the costs, because the primary objective of all investments is to improve overall organizational performance.

|  |  |  |
| --- | --- | --- |
| **Type** | **Potential Costs** | **Potential Benefits** |
| Quantitative | * Hardware/software upgrades * Fully-burdened cost of labor (salary + benefits) * Support costs for the application * Expected operational costs * Training costs for users to learn the application * Training costs to train developers in new/updated technologies | * Reduced operating costs * Reduced personnel costs from a reduction in staff * Increased revenue from additional sales of your organizations products/services |
| Qualitative | * Increased employee dissatisfaction from fear of change | * Improved decisions as the result of access to accurate and timely information * Raising of existing, or introduction of a new, barrier to entry within your industry to keep competition out of your market * Positive public perception that your organization is an innovator |

 The table includes both qualitative factors, costs or benefits that are subjective in nature, and quantitative factors, costs or benefits for which monetary values can easily be identified.  I will discuss the need to take both kinds of factors into account when performing a cost/benefit analysis.

**OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Not only must an application make economic and technical sense, it must also make operational sense.

|  |  |
| --- | --- |
| **Operations Issues** | **Support Issues** |
| * What tools are needed to support operations? * What skills will operators need to be trained in? * What processes need to be created and/or updated? * What documentation does operations need? | * What documentation will users be given? * What training will users be given? * How will change requests be managed? |

Very often you will need to improve the existing operations, maintenance, and support infrastructure to support the operation of the new application that you intend to develop.  To determine what the impact will be you will need to understand both the current operations and support infrastructure of your organization and the operations and support characteristics of your new application. To operate this application END-TO-END VMS.

**SELECTED SOFTWARE**

**IMPLEMENTATION ON (PYTHON):**

**What Is A Script?**

Up to this point, I have concentrated on the interactive programming capability of Python.  This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

**Scripts are reusable:**

Basically, a script is a text file containing the statements that comprise a Python program.  Once you have created the script, you can execute it over and over without having to retype it each time.

**Scripts are editable:**

Perhaps, more importantly, you can make  different versions of the script by modifying the statements from one file to the next using a text editor.  Then you can execute each of the individual versions.  In this way, it is easy to create different programs with a minimum amount of typing.

**You will need a text editor:**

Just about any text editor will suffice for creating Python script files.

You can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor if you want to.

**Python Libraries used in this project:**

**absl-py==0.9.0**

**appdirs==1.4.3**

**astor==0.8.1**

**certifi==2019.3.9**

**chardet==3.0.4**

**coreapi==2.3.3**

**coreschema==0.0.4**

**distlib==0.3.0**

**Django==2.2.13**

**django-cors-headers==2.4.0**

**djangorestframework==3.9.1**

**filelock==3.0.12**

**gast==0.2.2**

**google-pasta==0.2.0**

**grpcio==1.27.2**

**h5py==2.10.0**

**idna==2.9**

**importlib-metadata==1.6.0**

**importlib-resources==1.4.0**

**itypes==1.1.0**

**Jinja2==2.11.1**

**Keras==2.2.4**

**Keras-Applications==1.0.8**

**Keras-Preprocessing==1.1.0**

**Markdown==3.2.1**

**MarkupSafe==1.1.1**

**numpy==1.18.2**

**opt-einsum==3.2.1**

**Pillow==7.1.0**

**pipenv==2018.11.26**

**protobuf==3.11.3**

**pytz==2019.3**

**PyYAML==5.3.1**

**requests==2.23.0**

**scipy==1.4.1**

**six==1.14.0**

**sqlparse==0.3.1**

**tensorboard==1.15.0**

**tensorflow==1.15.2**

**tensorflow-estimator==1.15.1**

**termcolor==1.1.0**

**uritemplate==3.0.1**

**Fundamental concepts of Machine Learning domain**

In our system processing starts with Data collection, through some the pre-processing, feature extractor steps to be allowed and then finally classifies emotions.

**DATA COLLECTION**

**Dataset** : Statistical features derived from three electroencephalography (EEG) datasets are presented in visual space and processed in 2D and 3D space as pixels and voxels respectively. Three datasets are benchmarked, mental attention states and emotional valences from the four TP9, AF7, AF8 and TP10 10–20 electrodes and an eye state data from 64 electrodes. Seven hundred twenty-nine features are selected through three methods of selection in order to form 27 **×** 27 images and 9 **×** 9 **×** 9 cubes from the same datasets. CNNs engineered for the 2D and 3D preprocessing representations learn to convolve useful graphical features from the data. *Main results.* A 70/30 split method shows that the strongest methods for classification accuracy of feature selection are One Rule for attention state and Relative Entropy for emotional state both in 2D. In the eye state dataset 3D space is best, selected by Symmetrical Uncertainty. Finally, 10-fold cross validation is used to train best topologies. Final best 10-fold results are 97.03% for attention state (2D CNN), 98.4% for Emotional State (3D CNN), and 97.96% for Eye State (3D CNN). *Significance.* The findings of the framework presented by this work show that CNNs can successfully convolve useful features from a set of pre-computed statistical temporal features from raw emotional images. The high performance of K-fold validated algorithms argue that the features learnt by the CNNs hold useful knowledge for classification in addition to the pre-computed features.

**Processing Techniques**

The data was obtained from two individuals for three minutes per state as positive, neutral, and negative (1 male, 1 female). The TP9, AF7, AF8, and TP10 EEG sites with dry electrodes were registered with the headband from Muse EEG. For six minutes of resting neutral data, the stimuli used to extract emotions were also recorded: Negative: Marley and Me (Twentieth Century Fox), Negative: Up (Walt Disney Pictures), Negative: My Girl (Imagine Entertainment), Positive: La La Land (Summit Entertainment), Positive: Slow Life (BioQuest Studios), and Positive: Funny Dogs (Walt Disney Pictures) (MashupZone). To mathematically describe data in temporal space, a statistical extraction approach is used. The data is split into two sets; 70% as training set and 30% as testing set for all classifiers. This choice is made after testing other data splits which yielded lower signal classification accuracies.

**Feature Extraction**

Feature Extraction is the most important step which can be used to analyze and explore the image properly. It is a process reduces the number of features and creating new features from existing ones in dataset. Feature extraction is based on the ABCD rule, the ABCD stands for Asymmetry, Border structure, Color variation, and Diameter of that image. Atleast step use CNN model and classifies emotions like positive, negative and neutral.

**Classification**

Classification algorithms typically employ two phases of processing: training and testing data. Image classification is a set of target classes (objects to identify in images), and train a model to recognize them using labeled. Convolutional Neural Networks (CNN or ConvNet) is used for image recognition and classification.

**TESTING**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

**SYSTEM TESTING**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**MODULE TESTING**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**INTEGRATION TESTING**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested.

The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**Python Web Frameworks**

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

**Why are web frameworks useful?**

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format templating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

**Comparing web frameworks**

There is also a repository called [compare-python-web-frameworks](https://github.com/mattmakai/compare-python-web-frameworks) where the same web application is being coded with varying Python web frameworks, templating engines and object.

**Web framework resources**

* When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
* Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
* what is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.
* Django vs Flash vs Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
* This fascinating blog post takes a look at the  code complexity of several Python web frameworks by providing visualizations based on their code bases.
* Python’s web frameworks benchmarks  is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
* What web frameworks do you use and why are they awesome? is a language agnostic Reddit discussion on web frameworks. It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
* This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?". The votes aren't as important as the list of the many frameworks that are available to Python developers.

## Web frameworks learning checklist

1. Choose a major Python web framework (Django or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the [deployment](https://www.fullstackpython.com/deployment.html)section to make it accessible on the web

**TEST CASES:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | | **Test Steps** | | | **Test Case Status** | **Test Priority** |
|  | **Step** | | **Expected** | **Actual** |
| 01 | Upload the EEG images | Verify either file is loaded or not | | If dataset is not uploaded | It cannot display the file loaded message | File is loaded which displays task waiting time | High | High |
| 02 | Upload EEG images | Verify either dataset loaded or not | | If dataset is not uploaded | It cannot display dataset reading process completed | It can display dataset reading process completed | low | High |
| 03 | Preprocessing | Whether preprocessing on the dataset applied or not | | If not applied | It cannot  display the necessary data for further process | It can display the necessary data for further process | Medium | High |
| 04 | Prediction CNN model | Whether  Prediction algorithm applied on the data or not | | If not applied | CNN model is generated and predicted | Crack is detected using CNN model | High | High |
| 05 | Display classificati-on detection | Whether predicted emotions data is displayed or not | | If not displayed | It cannot view prediction containing EEG image data | It can view prediction containing EEG classification of emotional data | High | High |
| 06 | Noisy Records Chart | Whether the graph is displayed or not | | If graph is not displayed | It does not show the variations in between clean and noisy records | It shows the variations in between clean and noisy records | Low | Medium |

TABLE :TESTCASES

**SYSTEM REQUIREMENTS:**

**Hardware requirements**

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

* System : Pentium IV 2.4 GHz.
* Hard Disk : 100 GB.
* Monitor : 15 VGA Color.
* Mouse : Logitech.
* RAM : 1 GB.

**SOFTWARE REQUIREMENTS**

#### The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

#### Operating system : Windows 7/10

* Coding Language : Python
* Development Kit : Anaconda IDE
* Dataset : EEG Emotion dataset
* Front End : Html,CSS,JS
* Framework : Flask

**SYSTEM DESIGN**

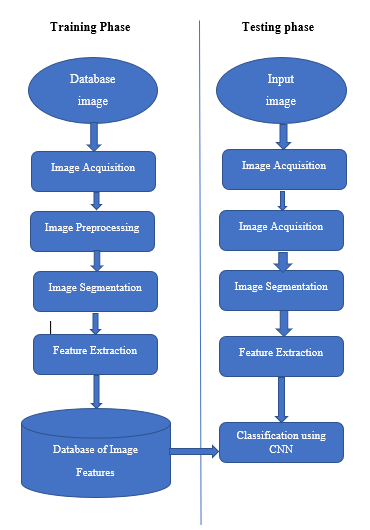
**Convolution Operation:** The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters.

* [**ReLU Layer**](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-1b-relu-layer/)**:** transform functions only activates a node if the input is above a certain quantity. While the data is below zero, the output is zero, but when the information rises above a threshold. It has a linear relationship with the dependent variable.

### **Pooling Layer:** In the layer, we shrink the image stack into a smaller size. Pooling is done after passing by the activation layer.

* **Fully Connected Layer:** Fully Connected layers in neural networks are those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are full connected layers which compiles the data extracted by previous layers to form the final output.
* **Softmax:** we use this function at last layer of neural network which calculates the probabilities distribution of the event over ’n’ different events. The main advantage of the function is able to handle multiple classes.

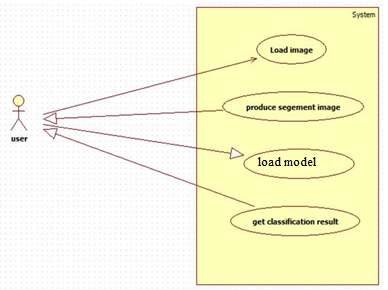
**Flow Chart:**



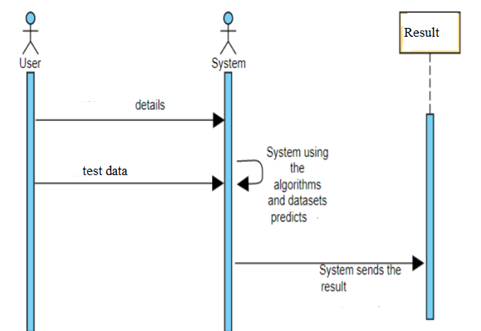
**Fig:Dataflow diagram**

In order to build a machine leaning model it consists of two phase namely testing and training phase were the model is first trained and an input is given to test the model which is called the test data. The model consists of several image processing steps such as image acquisition, image pre-processing, segmentation, feature extraction and CNN classifier to classify the diseases.

**Usecase**



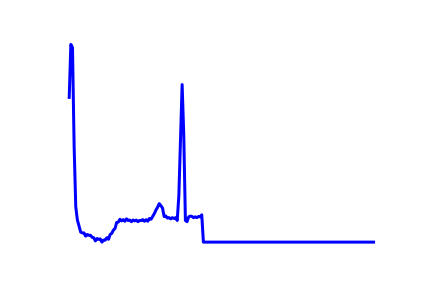
**Fig:Usecase**

**Sequence diagram**

**METHODOLOGY**

* Preprocessing and Training the model (CNN): The database is Preprocessed such as Image reshaping, resizing and conversion to an array form.
* , the data was obtained from two individuals for three minutes per state as positive, neutral, and negative (1 male, 1 female). The TP9, AF7, AF8, and TP10 EEG sites with dry electrodes were registered with the headband from Muse EEG. For six minutes of resting neutral data, the stimuli used to extract emotions were also recorded, and classifies as Positive, negative, neutral.
* The train database is used to train the model (CNN) so that it can identify the test image and classifies the emotion. CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, MaxPooling2D. After the model is trained successfully, the software can identifies the emotion as negative, positive, neutral.
* Database collection: Initial step for any image processing based project is acquiring proper database which is valid. Data available here is not labeled .So the first task is to clean and label the database. There is a huge database so basically the images with better resolution and angle are selected. After selection of images we should have deep knowledge about the different classification they have. Huge research is done from different EEG data repository. Different types of EEG signals are studied and corresponding. After detail study, labeling is done by segregating the images and with results emotions into Positive, Negative, Neutral.

**Implementation(Results and analysis) or Screenshots**



**CONCLUSION**

Research into emotional recognition has many fascinating pathways. This work presents image processing and analysis of emotional recognition. Finally, most emotion detection algorithms are applied to characterize various emotional states like positive, negative, neutral. Studies of the evolution of an emotional state are restricted and do not help to learn the changing processes of the emotions of a person. In future studies, we will explore how much virtual reality can help determine the present state of emotional behaviour through immersive experiences.

**FUTURE SCOPE**

Using new Different technologies and method we can make more faster an efficient application for user. The system presented in this project was able to perform accurately, however there are still a number of issues which need to be addressed. First of all, we consider only three states in this project therefore the scope of emotion classification is limited. In order to increase the scope of the emotion classification detection large datasets of different disease should be use.

**REFERENCE**

[1] H.J. Edenberg, Genetics of alcoholism, Science 282 (5392) (1998) 1265i–1265.

[2] S.A. Churchill, L. Farrell, Alcohol and depression: evidence from the 2014 health survey for england, Drug Alcohol Depend. 180 (2017) 86–92.

[3] W.H. Organization, Alcohol Key Facts, Technical Report, 2018. Genf, Schweiz.

[4] N. Ahmadi, Y. Pei, M. Pechenizkiy, Detection of alcoholism based on EEG signals and functional brain network features extraction, 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), IEEE, 2017.

[5] U.R. Acharya, V. S, S. Bhat, H. Adeli, A. Adeli, Computer-aided diagnosis of alcoholism-related EEG signals, Epilepsy Behav. 41 (2014) 257–263.

[6] M. Bhler, K. Mann, Alcohol and the human brain: a systematic review of different neuroimaging methods, Alcohol.: Clin. Exp. Res. 35 (10) (2011) 1771–1793.

[7] T.M. Nunes, A.L. Coelho, C.A. Lima, J.P. Papa, V.H.C. de Albuquerque, EEG Signal classification for epilepsy diagnosis via optimum path forest – a systematic assessment 136 (2014) 103–123.

[8] S.-H. Wang, K. Muhammad, J. Hong, A.K. Sangaiah, Y.-D. Zhang, Alcoholism identification via convolutional neural network based on parametric relu, dropout, and batch normalization, Neural Comput. Appl. (2018).

[9] E. Abdulhay, E. V., C. M., B. V.S., N. K., Automated diagnosis of epilepsy from EEG signals using ensemble learning approach, Pattern Recognit. Lett. (2017).

[10] D. Bansal, R. Mahajan, Chapter 1 - Introduction, in: D. Bansal, R. Mahajan (Eds.), EEG-Based Brain-Computer Interfaces, Academic Press, 2019, pp. 1–19