|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Headset | Drone | Directions | Programming Tools | Approach |
| 1[2] | Emotional Insight (5-channels) | Parrot Mambo Fly Drone | Up, down, left, right, forward and backward | Emotiv Cortex and python-3 | Facial expressions and mental commands |
| 2[5] | Emotiv EPOC (16-channels) | Quadcopter AR 2.0 | Left/right smile, frown and left/right wink | -- | Facial expressions |
| 3[6] | Emotiv COPD neuroheadset (16-channels) | LEDs were used | Up, down, left, right, forward and backward | Python | -- |
| 4[7] | Emotiv Inc. headset | Parrot 2 AR drone |  | Python | Motor imagery |
| 5[8] | 16 lead EEG headset | Parrot Mambo drone | Up, down, left, right, forward and backward | Python | Facial expressions |
| 6[9] | Emotiv EPOC helmet | Custom quadrotor drone | Up, down, left and right | Python and C# | Mental commands |
| 7[10] | Emotiv insight headset | Parrot Rolling spider drone | Up, down, left and right | -- | Mental Commands |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **System Development phases** | **Sep** | | **Oct** | | **Nov** | |
| **1-15** | **16-30** | **1-15** | **16-30** | **1-15** | **16-30** |
| **Data Recording** |  |  |  |  |  |  |
| **Pre-Processing** |  |  |  |  |  |  |
| **Model creation** |  |  |  |  |  |  |
| **Model Validation** |  |  |  |  |  |  |
| **Model Deployment** |  |  |  |  |  |  |
| **Command transfer** |  |  |  |  |  |  |

C# application nhn bany gi kyun k humen koi interface nhn banna blky suto krna hai

# 

# **7.0 System Architecture:**

The system will comprise of following phases:

Data recording

ML model creation

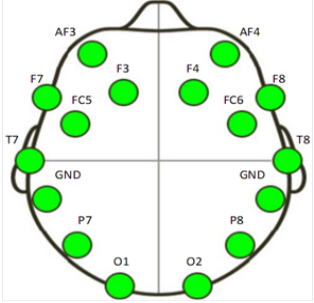
The proposed system will have the following architecture: starting with the headset, OpenBCI, will be used for brain signal acquisition from subject’s mind via scalp from different channel locations as shown in Figure 1 and will be transferred to computer via Bluetooth. The signal will be received by the OpenBCI-GUI that allows you to view your EEG data, interpret your results and creates a window into your brain like never before. We will build a desktop application in C# that will act as intermediator between brain and drone. This application will be accountable for receiving raw signals from OpenBCI software, applying preprocessing on the signals and will use ML model for signal classification.

Figure 1 Location of the sixteen channels by OpenBCI

The ML model is the "object" that remains after a machine learning algorithm is applied to training data, and it contains the rules, numbers, and other algorithm-specific data structures that are required to create predictions. The desktop application will generate an output command that will act as an input for the drone and these signals will be transferred to drone via wi-fi. The drone we will be using is Tello EDU. The Drone is programmed in python and can be attached to desktop, laptop or remote controller using the Wi-Fi technology. The proposed system architecture is shown in Figure 2.

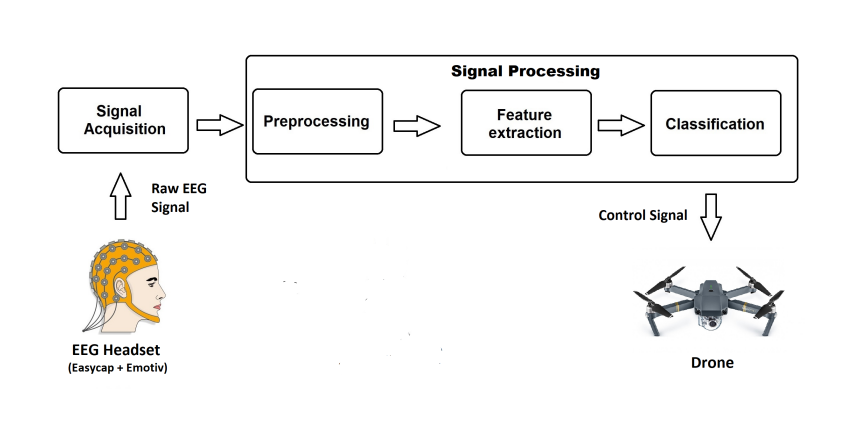


Figure 2 Proposed System Architecture