|  |
| --- |
|  |
| Digital Twin in Healthcare |
| |  |  |  | | --- | --- | --- | |  | 2/28/23 | Graduation Project | |

A picture containing text

Description automatically generated

**Faculty of Computers and Artificial Intelligence**

**Cairo University**

**Digital Twinning in HealthCare**

Supervised by:

**Dr. Aboul-Ella Otifey**

**Dr. Eman Ahmed Elsayed**

**Presented By:**

|  |  |
| --- | --- |
| **NAME** | **ID** |
| **Manar Emad ELdin Hassan Ahmed** | **20190552** |
| **Abdelrahman Mahmoud Alsayed Ibrahim** | **20190732** |
| **Salma Magdi Fahmy** | **20190245** |
| **Mahmoud Kamal Ahmed** | **20190501** |
| **Baraa Alfaris Mohammed** | **20190777** |
| **Mohamed Ashraf Hassan** | **20190421** |

**Table of Contents**

[1) Introduction 5](#_Toc128428868)

[1.1 Scope 6](#_Toc128428869)

[1.2 Objective 6](#_Toc128428870)

[1.3 Problem Definition 6](#_Toc128428871)

[1.4 Related Work 7](#_Toc128428872)

[1.4.1 Robotic Arm 7](#_Toc128428873)

[2) Background 10](#_Toc128428874)

[2.1 BCI 10](#_Toc128428875)

[2.2 EEG Signal 11](#_Toc128428876)

[2.3 Digital Twin Technology 13](#_Toc128428877)

[2.3.1 What is Digital Twin? 13](#_Toc128428878)

[2.3.2 How does this Technology Work? 13](#_Toc128428879)

[2.3.3 What Challenges has it Solved? 14](#_Toc128428880)

[2.3.4 Who Invented it? 14](#_Toc128428881)

[2.3.5 When to Use it? 15](#_Toc128428882)

[2.3.6 Why and How to Design it? 15](#_Toc128428883)

[2.3.7 Benefits 16](#_Toc128428884)

[2.3.8 Examples 17](#_Toc128428885)

[2.3.9 How has it Impacted the Industry? 17](#_Toc128428886)

[2.3.10 Where is it Used? 17](#_Toc128428887)

[2.4 CNN Model 18](#_Toc128428888)

[2.4.1 Introduction to CNN 18](#_Toc128428889)

[2.4.2 Components of CNN 19](#_Toc128428890)

[2.5 Blender & BPY 23](#_Toc128428891)

[2.5.1 What is Blender? 23](#_Toc128428892)

[2.5.2 Features & Functions 23](#_Toc128428893)

[2.5.3 BPY (Blender API) 25](#_Toc128428894)

[2.6 Dataset 26](#_Toc128428895)

[2.7.1 Overview 26](#_Toc128428896)

[2.7.2 Dataset Description 27](#_Toc128428897)

[3) Proposed Model 28](#_Toc128428898)

[3.1 Design 28](#_Toc128428899)

[3.2 Preprocessing 29](#_Toc128428900)

[3.3 Feature Extraction 30](#_Toc128428901)

[3.4 EEGNET 31](#_Toc128428902)

[3.5 TCN 33](#_Toc128428903)

[3.6 LSTM 34](#_Toc128428904)

[4) Results 35](#_Toc128428905)

[4.1 Simulation 35](#_Toc128428906)

[4.2 Project Progress 38](#_Toc128428907)

# **Introduction**

Previously, the cutting-edge prosthetics were only available to military veterans injured in service. Where Prosthetics offered previously were basic models, with limited open and close gripping motions, or others were cosmetic with no function.

Eligible patients must have enough residual upper arm muscles to send **signals** that create intuitive movements, but children as young as nine can use them, allowing them to improve their play and learning.

Each patient is carefully assessed to find the right type of prosthetic for them depending on their requirements and capabilities.

Where these incredible multi-grip prosthetics have already made a huge difference to veterans and so it is fantastic to be able to offer them to all patients who need them.

The arms, for both children and adults, use the very latest tech which will boost peoples’ independence and change the lives of dozens across the country.

It will massively change peoples’ lives because they will be able to do things more independently – they have amazing functionality. Where they can hold a paint brush and paint or pick up a glass and drink from it.

So, what if we could move these arms using our **EEG** signal**?**

*The amputees nowadays usually wear a prosthetic arm which are not functionable. These prosthetic arms unable to help them to carry out their basic daily work independently. However, the EEG based brain wave controlled intelligent prosthetic arm is able to assist the amputees to carry out basic daily work. By using the EEG brain wave controlled, the amputees can control the movement of the prosthetic arm by visualizing the movement of the arm. Besides the amputees, the paralyzed person can also use their thought to control the prosthetic arm. The EEG based brain wave controlled intelligent 3D printed prosthetic arm is designed to use the amputee's brain wave to control the movement of the prosthetic arm. The amputee needs to be trained in order to control every arm movement correctly. The EEG signal needs to be collected from the amputee on visualizing on 4 left forearm movements which are up, down, left and right. After that, signal pre-processing and Discrete Wavelet Transform (DWT) is performed on the EEG signal to obtain the alpha wave. Then, FFT is performed to the alpha wave to convert it to frequency-domain.*

## **Scope**

In applying **Digital Twins** to healthcare we extrapolate from the typical approach used manufacturing and illustrate with a simple example of a car battery. Here multi-spectral data is collected that will assist to determine the life of the battery so that the mechanic decision maker can decide when is prudent and opita to replace the battery. Translating this to the healthcare domain with a disability patient we similarly collect multi-spectral healthcare data that is relevant and pertinent for the clinician decision maker to assess the best treatment protocol and survivorship of the presenting patient. Clearly, the healthcare system is more complex than that of a car battery especially as healthcare deals with biological systems that are messy and complex; however, the underlying principles are similar.

## **Objective**

Modern robotic systems can describe, manipulate, and depict the behavior of a real object using **digital twin** technology. Currently, the industry plans to use this technology extensively. Making a dynamic model of the object for the simulation software is one of the crucial responsibilities in this domain. This outlines the fundamental ideas behind the manipulator's construction using digital twins, taking into account both its dynamic and mass-dimensional properties. It has been demonstrated that the best way to ensure the effectiveness of technological activities, as well as to set standard tolerance grades and restrict deviations within IT, is to apply polynomial algorithms in the control system.

**We therefore focus on using digital twinning to design a prosthetic smart arm using a patient's EEG signal.**

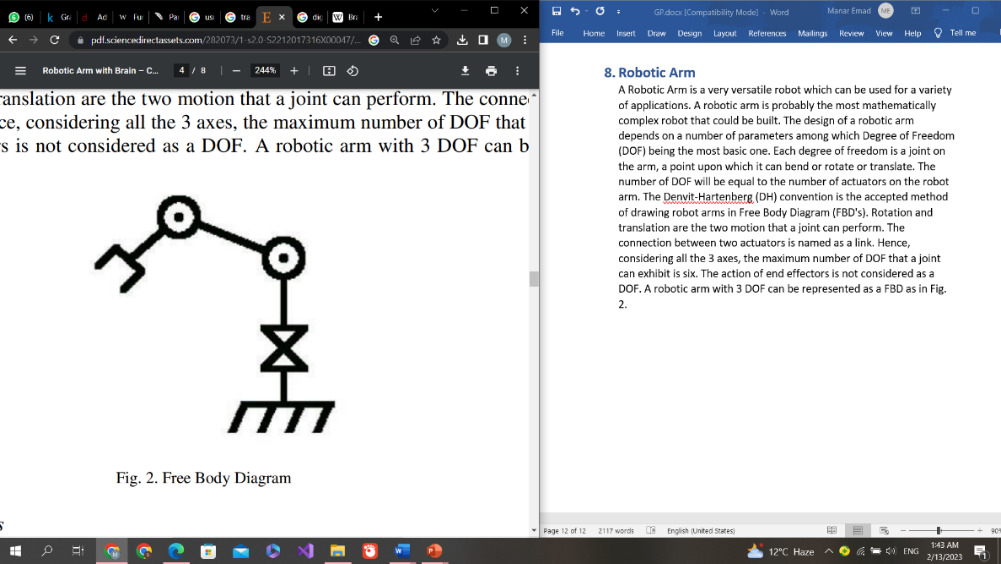
## **Problem Definition**

Design aprosthetic smart arm using a patient's EEG signal and implement this arm in virtual world to monitor arm interactions would be taken in different cases to avoid bad scenarios that could be happened to help disabled people.

## **Related Work**

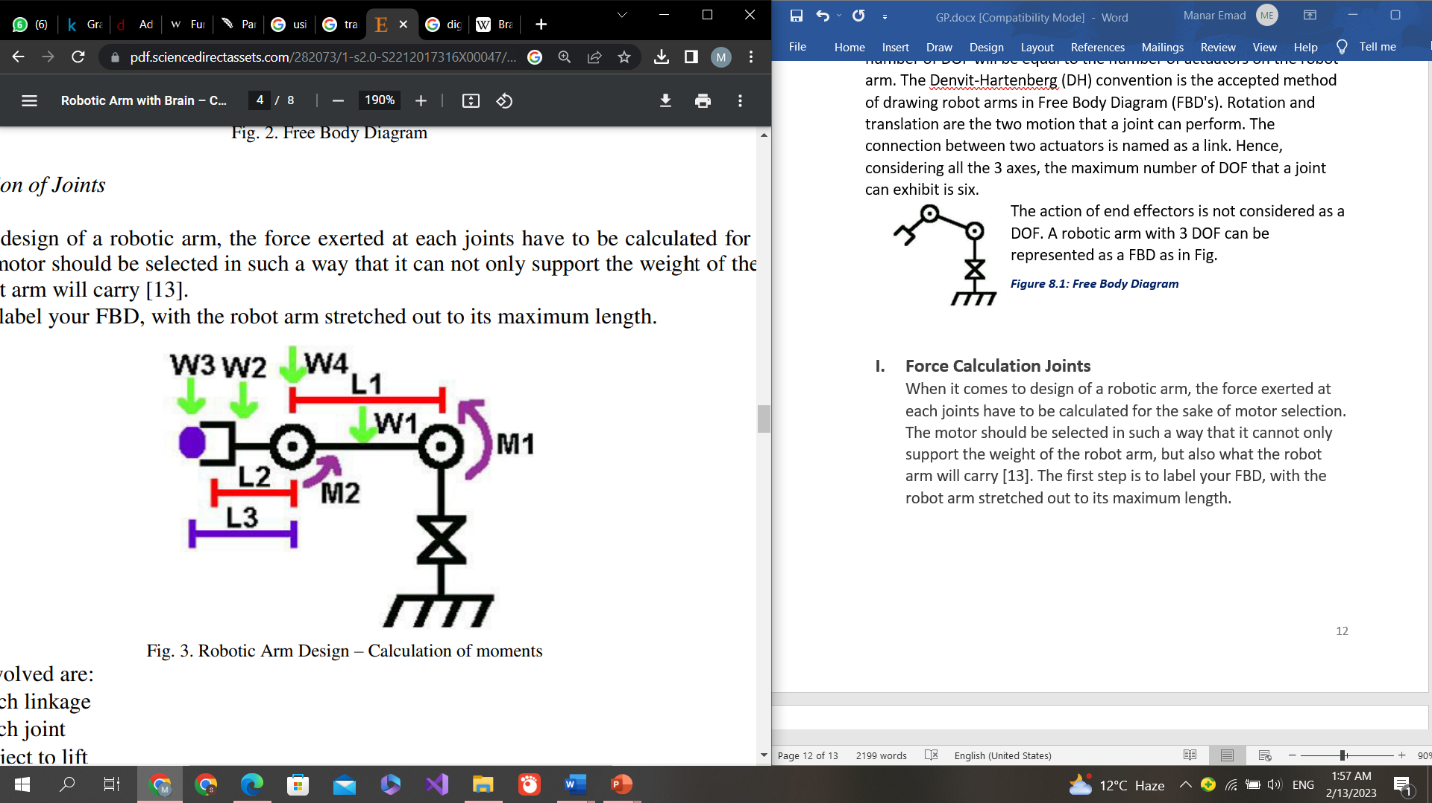
### **Robotic Arm**

A Robotic Arm is a very versatile robot which can be used for a variety of applications. A robotic arm is probably the most mathematically complex robot that could be built. The design of a robotic arm depends on a number of parameters among which Degree of Freedom (DOF) being the most basic one. Each degree of freedom is a joint on the arm, a point upon which it can bend or rotate or translate. The number of DOF will be equal to the number of actuators on the robot arm. The Denvit-Hartenberg (DH) convention is the accepted method of drawing robot arms in Free Body Diagram (FBD's). Rotation and translation are the two motion that a joint can perform. The connection between two actuators is named as a link. Hence, considering all the 3 axes, the maximum number of DOF that a joint can exhibit is six.

The action of end effectors is not considered as a DOF. A robotic arm with 3 DOF can be represented as a FBD as in Fig.

***Figure ‎1.4.1: Free Body Diagram***

#### **Force Calculations of Joints**

When it comes to design of a robotic arm, the force exerted at each joints have to be calculated for the sake of motor selection. The motor should be selected in such a way that it cannot only support the weight of the robot arm, but also what the robot arm will carry. The first step is to label your FBD, with the robot arm stretched out to its maximum length.

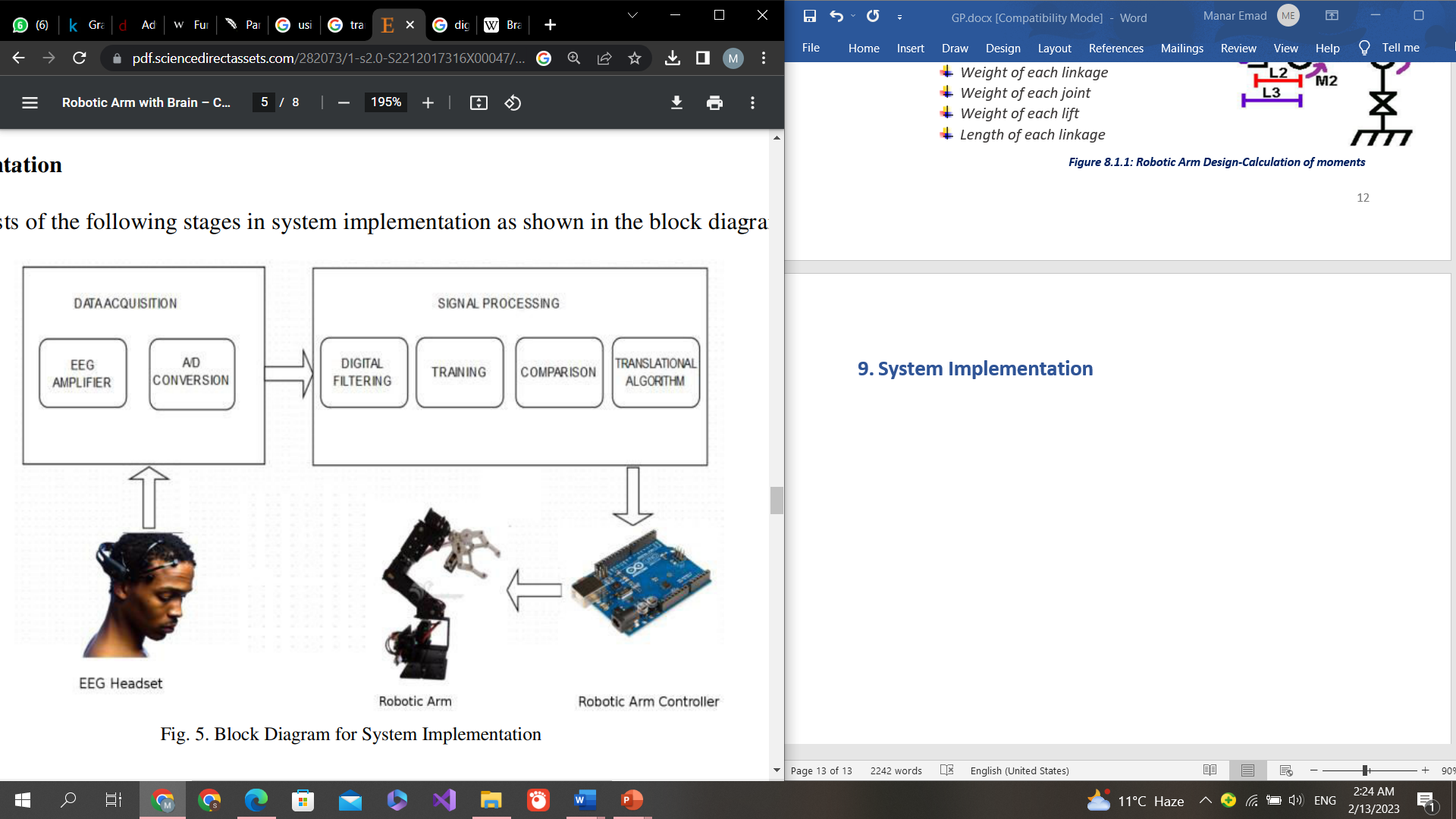
**The parameters involved are**:

* *Weight of each linkage*
* *Weight of each joint*
* *Weight of each lift*
* *Length of each linkage*

***Figure ‎1.4.1.1: Robotic Arm Design-Calculation of moments***

#### **System Implementation**

The system consists of the following stages in system implementation as shown in the block diagram.



***Figure ‎1.4.1.2: Block Diagram for System Implementation***

* 1. **Signal Acquisition**

In the signal acquisition part of BCI operation, the chosen input is:

* *Acquired by the recording electrodes*
* *Amplified*
* *Digitized*
  1. **Digital Filtering**

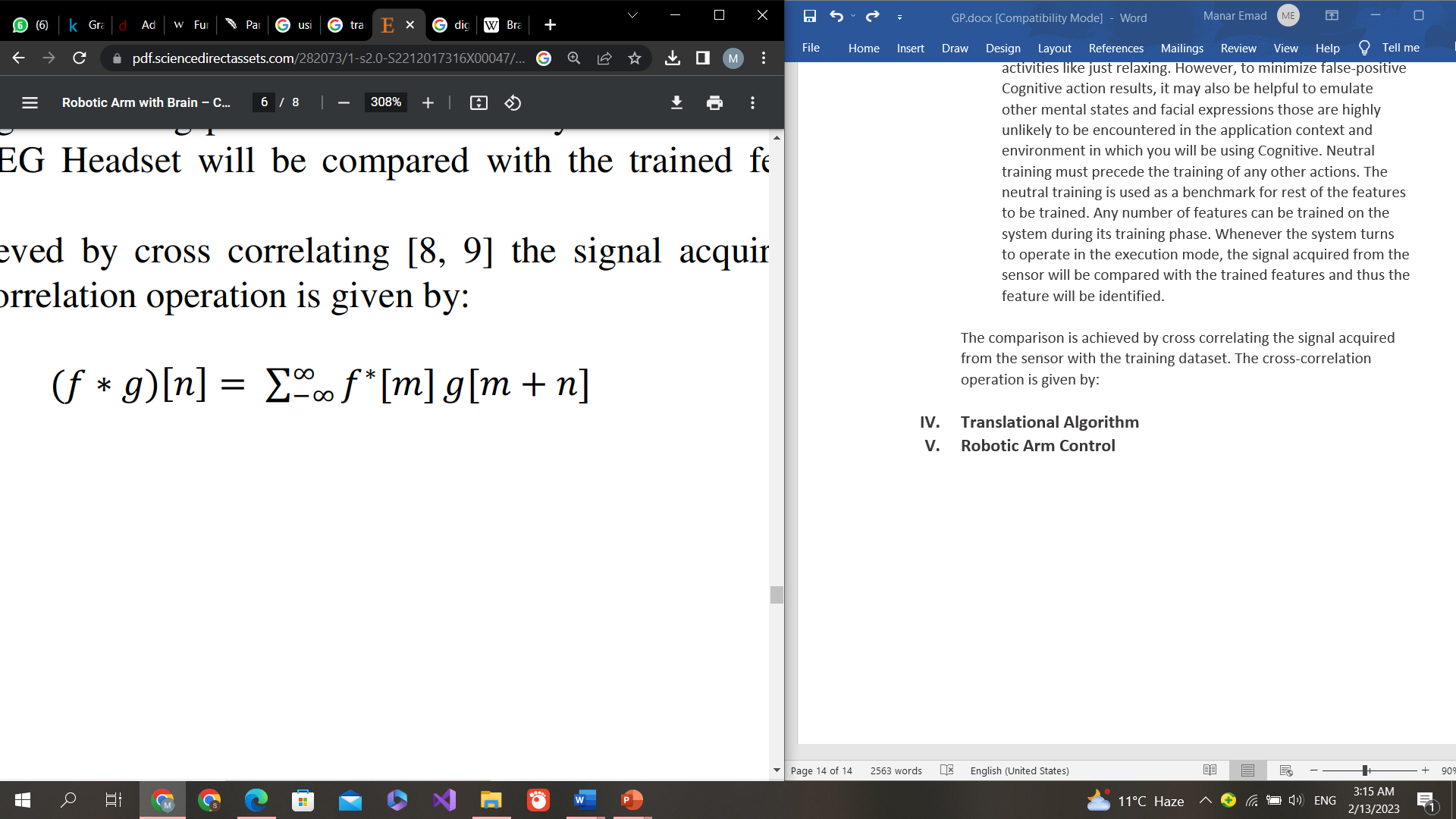
To filter the EEG signal there are different ways. One of them using notch filters, these notch filters remove the noise if any caused by the power supply. Where power supply usually causes serious disfiguration to the EEG signal acquired which is overcome.

* 1. **Training Comparison**

The system is trained based on the signals acquired from the sensors. The EEG signals corresponding to the various thoughts for movements is recorded and analyzed it.

* **Training Neutral**: The Neutral action refers to the user’s idle state of mind; which is not associated with any of the selected Cognitive actions. Typically, this means engaging in idle mental activities like just relaxing. However, to minimize false-positive Cognitive action results, it may also be helpful to emulate other mental states and facial expressions those are highly unlikely to be encountered in the application context and environment in which you will be using Cognitive. Neutral training must precede the training of any other actions. The neutral training is used as a benchmark for rest of the features.

The comparison is achieved by cross correlating the signal acquired from the sensor with the training dataset. The cross-correlation operation is given by:



* 1. **Translational Algorithm**

The translational algorithm is the one integral part of the whole system which generates device commands from the processed EEG signals. Commands thus generated by the translational algorithm help carrying out the user's intent.

* 1. **Robotic Arm Control**

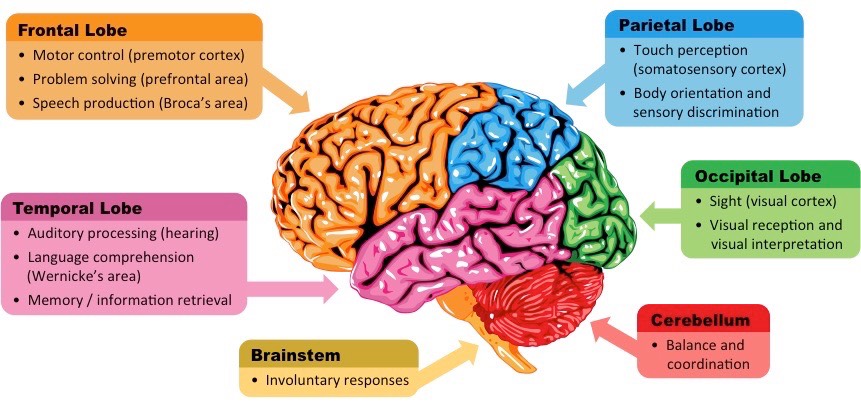
The output from the signal processing and the translational algorithm stage is in virtual reality of robotic arm, a real-life system with the help of interfacing software.

# **Background**

## **BCI**

An external device, most frequently a computer or robotic limb, and the electrical activity of the brain can directly communicate through a brain-computer interface (BCI), also known as a brain-machine interface (BMI) or smart brain. BCIs are frequently used to study, map, support, improve, or restore human cognitive or sensory-motor processes. Based on how close electrodes are placed to brain tissue, BCI implementations range from non-invasive (EEG, MEG, EOG, MRI) to partially invasive (ECoG and endovascular) to invasive (microelectrode array).

Because of the brain's cortical plasticity, signals from implanted prostheses can be processed by the brain like natural sensor or effector channels after adaption. The first neuroprosthetic implants in humans first arrived in the mid-1990s after many years of animal testing.

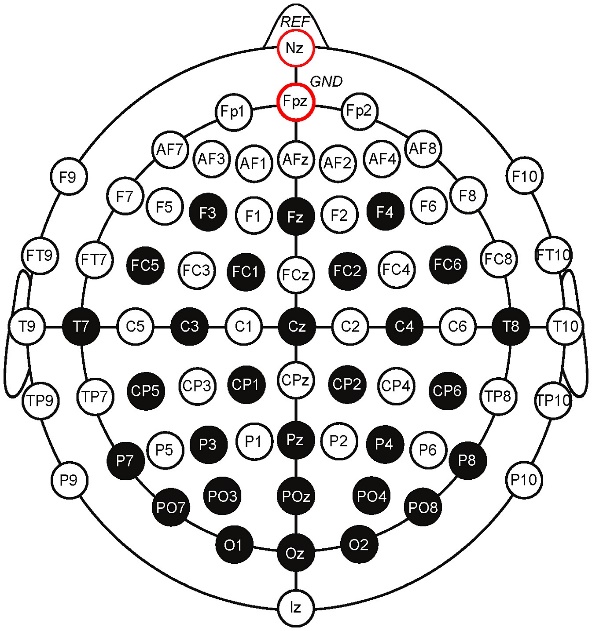


***Figure ‎2.1: Brain of Human***

Recent studies in human-computer interaction have demonstrated high levels of success in autonomous recognition of fall detection as a medical alarm, mental state (Relaxed, Neutral, Concentrating), mental emotional state (Negative, Neutral, Positive), and thalamocortical dysrhythmia through the application of machine learning to statistical temporal features extracted from the frontal lobe (EEG brainwave) data.

## **EEG Signal**

After signal conditioning and processing, electroencephalography (EEG), which uses sensors arranged over the scalp to measure and record electric impulses, can be utilized for robotics, prosthetic devices, applications in combat, gaming, and virtual reality.



***Figure ‎2.2.1: EEG Brain Sensor***

The goal of this work, which is solely based on brain-computer interface technology, is to actuate a robotic arm using device commands that are deduced from EEG data. This system, unlike any other on the market, is completely non-invasive, economical, and unique in that it can meet a variety of needs, including prosthetics. This study makes a low-cost system implementation recommendation that could possibly be a solid replacement for current prosthetic technologies like BIONICS.

Brain signals are low-amplitude, low-frequency signals that are difficult to acquire and process because they are so sensitive to environmental disturbances like noise. Brain signals are generally collected, amplified, and digitalized before being processed and examined by a personal computer.

In this study, we propose a method for creating a fully functional robotic arm that basically collects EEG signals from the human brain using an EEG headset, which are then converted to digital format, analyzed, and filtered. The end-effector and three degrees of freedom are operated on using the processed data from the raw EEG signals.

**The entire system consists of three:**

1. *Biomedical Phase*
2. *Signal Conditioning and Processing Phase*
3. *Hardware System Development Phase*

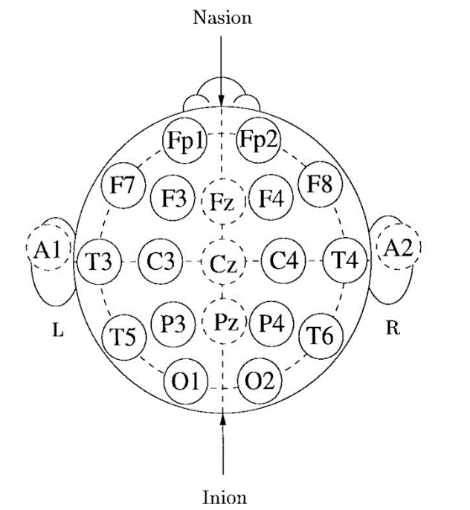
The electrode placement guidelines that must be followed in order to get an EEG signal are specified by the EEG lead system. A globally accepted technique for describing and applying the placement of scalp electrodes in the context of EEG signal capture is the International 10-20 electrode placement system. The 10/20 system was created to guarantee consistent repeatability, allowing for ongoing research on the topic across time as well as comparisons between various groups. This system is based on the correlation between an electrode's position and the cerebral cortex region beneath it. The numbers "10" and "20" indicate that the actual spacing between consecutive electrodes is 10% or 20% of the skull's overall front-to-back or right-to-left distance, respectively.

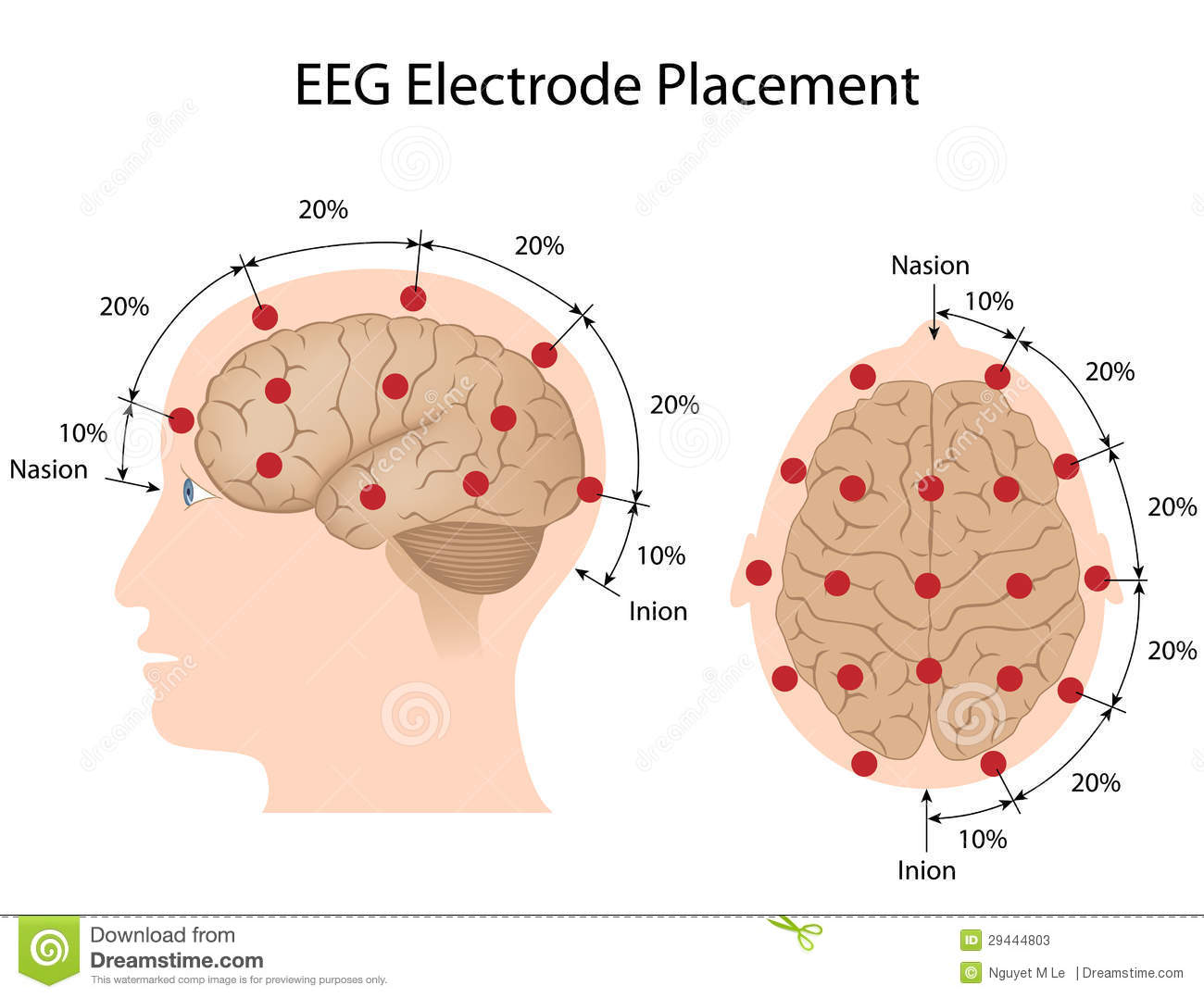
Each electrode implantation site is designated by a letter designating the lobe and a number designating the location in the hemisphere. Frontal, temporal, central, parietal, and occipital lobes are represented by the letters F, T, C, P, and O, respectively. The "C" letter is only used for identification since there is no central lobe in biology. The letter "z" stands for an electrode that is positioned on the midline of the human brain. The electrode locations on the right hemisphere are designated by even numbers (2,4,6,8) and the electrode placements on the left hemisphere by odd numbers (1,3,5,7). Additionally, A, Pg, and Fp stand in for the nasopharyngeal, frontal, and earlobe locations, respectively.

Two anatomical landmarks act as the reference to the whole electrode placement system.

The nasion is the point just above the bridge of the nose and inion, which is the lowest point of the skull from the back of the head.

The 10 20 electrode placement system is represented in Fig 2.2.2.





***Figure ‎2.2.3: EEG Lead System***

***Figure ‎2.2.2: 10 20 System***

## **Digital Twin Technology**

### **What is Digital Twin?**

A digital twin is a representation of a physical product, procedure, or service in the digital world. A digital twin is a digital representation of a real-world object, such as a jet engine, wind farm, or even larger objects like a building or even an entire city.

The digital twin technology can be used to duplicate processes in order to gather data and forecast their performance, in addition to physical assets.

In essence, a digital twin is a computer software that simulates how a process or product would work using data from the real world. To improve the results, these systems can incorporate software analytics, artificial intelligence, and the internet of things.

These virtual models have become a mainstay in contemporary engineering to spur innovation and boost efficiency thanks to the development of machine learning and elements like big data.

To put it briefly, developing one can enable the advancement of major technological trends, prevent expensive breakdowns in physical items, and test processes and services utilizing enhanced analytical, monitoring, and predictive skills.

### **How does this Technology Work?**

In order to create a mathematical model that simulates the original, professionals in applied mathematics or data science first study the physics and operational data of a physical object or system.

The designers of digital twins make sure that sensors that collect data from the physical counterpart may provide input to the virtual computer model. As a result, it is possible to duplicate and simulate what is happening with the original version using the digital version in real time, providing opportunity to learn more about performance and any potential issues.

With varying amounts of data dictating how closely the model matches the real-world physical version, a digital twin can be as complicated or as simple as you need.

The twin can be used with a prototype to provide input on the design or it can stand alone as a prototype to simulate what might happen when a built-in version is used.

### **What Challenges has it Solved?**

It has previously been utilized to address a great deal of problems because it can be used to a wide range of industries, including healthcare, automotive, and power generation. These difficulties include the need for improvements in race car efficiency as well as fatigue testing and corrosion resistance for offshore wind turbines. Hospital work flows and staffing have been modelled in other applications to identify process improvements.

A digital twin enables users to look at options for lengthening the lifecycle of a product, streamlining production processes, improving product development, and prototyping. In certain situations, a digital twin can visually depict an issue, allowing a solution to be developed and evaluated in the computer algorithm rather than in the real world.

### **Who Invented it?**

David Gelernter's 1991 book "Mirror Worlds" introduced the idea of digital twins; Michael Grieves of the Florida Institute of Technology later applied the idea to manufacturing.

Grieves officially proposed the idea of a digital twin in 2002 at a Society of Manufacturing Engineers conference in Troy, Michigan after relocating to the University of Michigan.

However, NASA was the organization that adopted the digital twin idea initially, and John Vickers from NASA gave the idea its name in a Roadmap Report from 2010. The concept was applied to the development of computerized spacecraft and capsule simulators for testing.

When Gartner listed the "digital twin" as one of the top 10 strategic technology trends for 2017, the idea of the digital twin gained even more traction. Since then, a wide range of industrial applications and procedures have made use of the notion.

### **When to Use it?**

Digital twin can be broken down into three broad types, which show the different times when the process can be used:

* **Digital Twin Prototype (DTP)** - This is undertaken before a physical product is created
* **Digital Twin Instance (DTI)** – This is done once a product is manufactured in order to run tests on different usage scenarios
* **Digital Twin Aggregate (DTA)** – This gathers DTI information to determine the capabilities of a product, run prognostics and test operating parameters

These broad kinds can be applied to a range of tasks, including as logistical planning, product development and redesign, quality control and management, and systems planning.

Whenever a product or process needs to be tested, whether for design, implementation, monitoring, or improvement, a digital twin can be used to save time and money.

### **Why and How to Design it?**

As was already noted, digital twins can be developed for a variety of purposes, including as testing a prototype or design, determining and tracking lifecycles, and determining how a product or process would function under various conditions.

By gathering information and building computer models to evaluate it, a digital twin design is developed. This may involve a real-time data and feedback interaction between the digital model and the actual physical thing.

#### **Data**

To develop a virtual model that can simulate the behaviors or states of the real-world object or procedure, a digital twin needs information about the relevant object or process. These details, which could be related to a product's lifecycle, could include engineering details, production procedures, or design specifications. Additionally, it could contain data about tools, supplies, components, processes, and quality assurance. Operation-related data can also be present, including real-time feedback, historical analysis, and maintenance logs. Business data or end-of-life techniques are two more types of data that can be employed in digital twin design.

#### **Modelling**

After the information has been acquired, computational analytical models can be built to depict operating effects, forecast states like weariness, and identify behaviors. These models can suggest actions based on engineering simulations, physical laws, chemical laws, statistics, machine learning, artificial intelligence, business logic, or goals. To help people understand the results, these models can be visualized using 3D representations and augmented reality modelling.

#### **Linking**

Digital twin discoveries can be combined to build an overview, for example, by including equipment twin findings into a production line twin, which in turn can inform a factory-scale digital twin. It is conceivable to enable smart industrial applications for actual world operational breakthroughs and enhancements by leveraging linked digital twins in this way.

### **Benefits**

Depending on how and where it is used, digital twins have different advantages. An oil pipeline or wind turbine, for instance, can be monitored using a digital twin to minimize maintenance requirements and associated expenses by millions of dollars. Additionally, digital twins can be used to prototype products before they are manufactured, lowering product faults and speeding up the time to market. Other applications of the digital twin can be found in process improvements, such as tracking workforce levels against production or coordinating a supply chain with needs for manufacturing or maintenance.

Increasing dependability and availability while enhancing performance through monitoring and simulation are common advantages. Additionally, they can save maintenance costs by foreseeing failure before it happens, decrease the risk of accidents and unexpected downtime due to failure, and ensure that production targets are not affected by scheduling maintenance, repair, and the procurement of new parts. By examining customization models, digital twins can also provide ongoing enhancements and guarantee product quality through real-time performance testing.

Despite all of its advantages, digital twins are not always appropriate because they might add to complexity. Some business issues can be resolved without the added effort and money spent on creating a digital twin.

### **Examples**

Examples of digital twin can be found across industry and beyond for manufacturing, maintenance and failure prevention / lifecycle monitoring.

Applications range from automotive uses where telemetry sensors provide feedback from vehicles to the digital twin program, factories where processes are simulated by digital twin to provide improvements, and healthcare where sensors can inform a digital twin to monitor and predict the well-being of a patient.

### **How has it Impacted the Industry?**

A digital twin is a simulation model that is created by combining data with technologies like artificial intelligence, machine learning, and software analytics. This model can be updated simultaneously with or instead of its physical counterpart. Because of this, businesses are able to evaluate a fully computerized development cycle, from design to deployment to decommissioning.

A digital twin enables industry to plan for downtime, respond to changing conditions, test design advancements, and much more by simulating actual assets, frameworks, and processes to create continuous data.

The development of Industry relies on the digital twin to enable automation, data interchange, and joined-up production processes as well as to reduce risk in product launch. Employees in the industry may keep an eye on processes in real time, offering early warnings of potential breakdowns and enabling real-time performance optimization and assessment with no loss of productivity.

### **Where is it Used?**

Digital twins are used in a wide variety of industries for a range of applications and purposes. Some notable examples include:

#### **HealthCare**

The medical sector has benefitted from digital twin in areas such as organ donation, surgery training, de-risking of procedures and **Prosthetics industry**. Systems have also modelled the flow of people through hospitals and track where infections may exist and who may be in danger through contact.

## **CNN Model**

### **Introduction to CNN**

A Deep Learning technique specifically created for working with images and videos is the convolutional neural network. It uses photographs as inputs, extracts and learns the image's attributes, then categorizes the images using the learnt features.

This programmed takes its cues from how the Visual Cortex functions in the human brain. Processing of visual data from the outside world is carried out by the visual cortex, a region of the human brain. It comprises a number of layers, and each layer functions independently, extracting different information from images or other visuals. Once all the information from each layer has been combined, the picture or visual is then evaluated or classed.

Similar to this, CNN uses a variety of filters that each collect data from the image, such as edges and various forms (vertical, horizontal, and round), which are then combined to identify the image.

Now, the question that may arise is: Why can't we get the same results with artificial neural networks? This is due to various negative aspects of ANN:

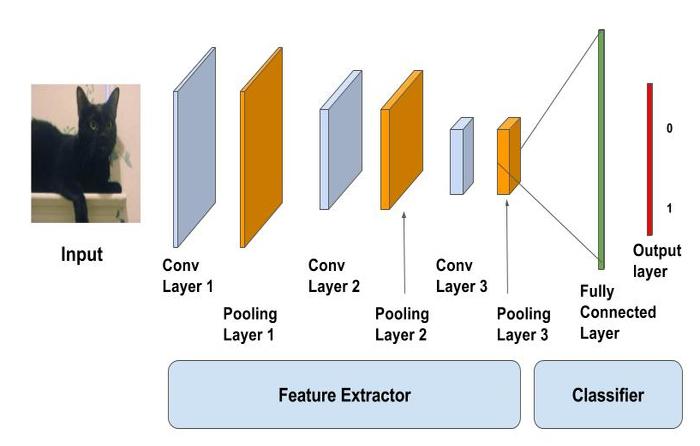
* + The amount of computing required to train an ANN model on huge images and various image channels is too great.
  + The second drawback is that, in contrast to a CNN model, it is unable to collect all of the information from an image, including its spatial dependencies.
  + Another issue is that the placement of the object in the image is important to ANN; if the location or site of the same object varies, the classification will be incorrect.

### **Components of CNN**

The CNN model works in two steps: feature extraction and Classification

The process of extracting information and features from images is called feature extraction, and once that process is complete, the images are passed on to the classification phase, where they are categorised according to the problem's target variable.

**A typical CNN model looks like this:**

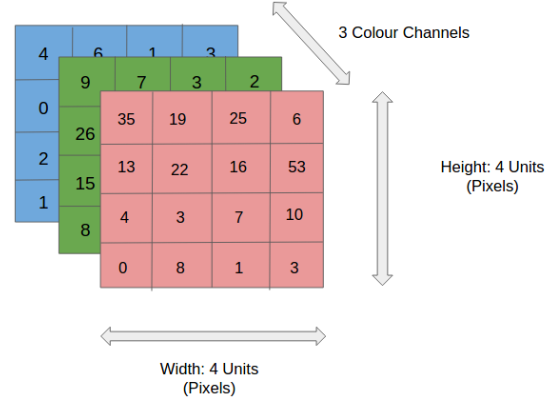


***Figure ‎2.4.2: CNN Model***

#### **Input Layer**

Our input image, as the name implies, can be either RGB or Grayscale. Each image is composed of pixels with values between 0 and 255. Before sending them to the model, we must normalise them, or transform the 0 to 1 range.

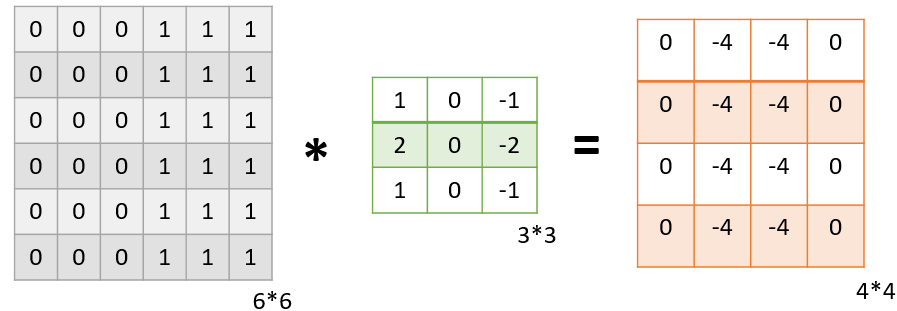
Below is the example of an input image of size 4\*4 and has 3 channels i.e RGB and pixel values.



***Figure ‎2.4.2.1: Input of Model***

#### **Convolution Layer + Activation Function**

The filter used to extract or detect the features of our input image is applied to it in the convolution layer. Several filtering operations are performed on the image to produce a feature map that aids in categorizing the input image. Let's use an illustration to better grasp this. We will use a 2D input image with normalized pixels for simplicity's sake.

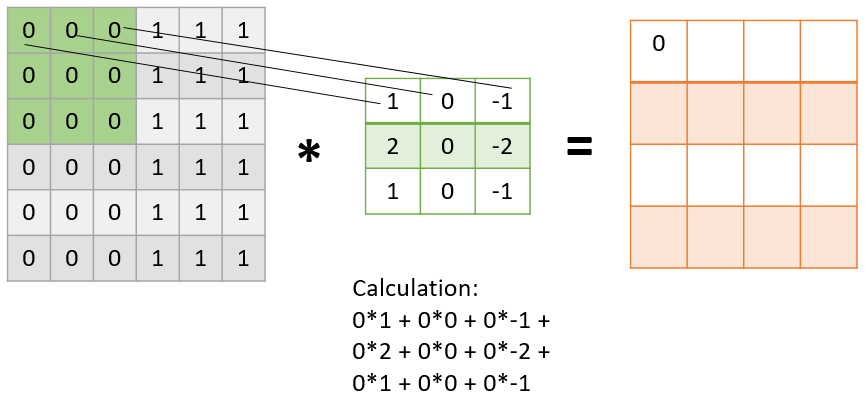
***Figure ‎2.4.2.2.I: Convolution Layer*** 

In the illustration above, we used a 3x3 filter to identify some features in an input image that was 6x6 in size. While we only used one filter in this example, in reality, many of these kinds of filters are used to extract data from images.

We obtain a 4\*4 Feature Map with some information about the input image as a result of applying the filter to the image. Such feature maps are frequently produced in real-world applications.

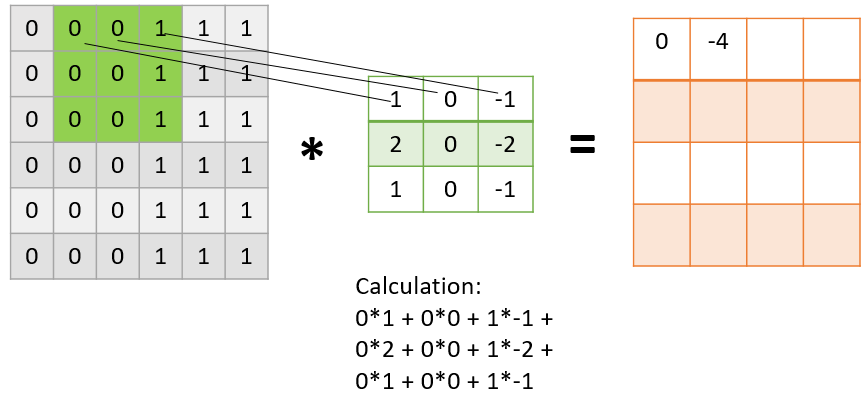
Let’s get into some maths behind getting the feature map in the above image.

***Figure ‎2.4.2.2.II: CNN Calculations1***



As presented in the above figure, in the first step the filter is applied to the green highlighted part of the image, and the pixel values of the image are multiplied with the values of the filter (as shown in the figure using lines) and then summed up to get the final value.

In the next step, the filter is shifted by one column as shown in the below figure. This jump to the next column or row is known as stride and in this example, we are taking a stride of 1 which means we are shifting by one column.

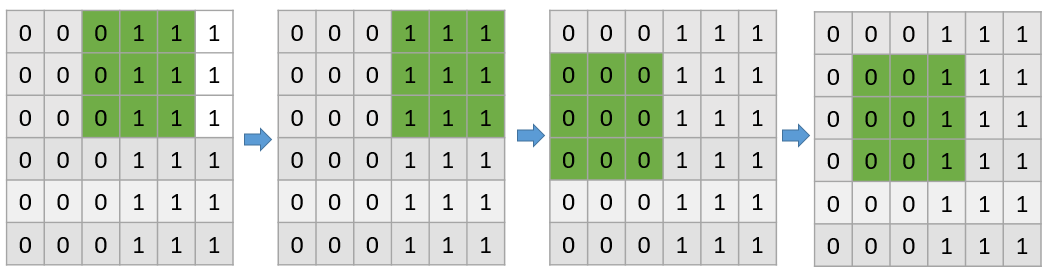


***Figure ‎2.4.2.2.III: CNN Calculations2***

The filter similarly covers the entire image, giving us our final Feature Map. Once the feature map has been obtained, nonlinearity is added by applying an activation function to it.

It's important to note that the size of our image is larger than the feature map we receive in this case. The size of the feature map reduces when the amount of stride is raised.

***Figure ‎2.4.2.2.IIII: Passing filter into img***



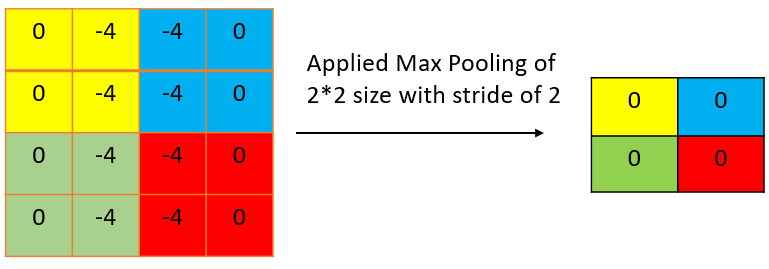
#### **Pooling Layer**

Following the convolutional layer, the pooling layer is used to shrink the feature map's size, aiding in the preservation of the input image's key details or features while speeding up computation.

***Using pooling, a lower resolution version of input is created that still contains the large or important elements of the input image.***

The most common types of Pooling are Max Pooling and Average Pooling. The below figure shows how Max Pooling works. Using the Feature map which we got from the above example to apply Pooling. Here we are using a **Pooling layer of size 2\*2 with a stride of 2**.

The maximum value from each highlighted area is taken and a **new version of the input image is obtained which is of size 2\*2 so after applying Pooling the dimension of the feature map has reduced.**

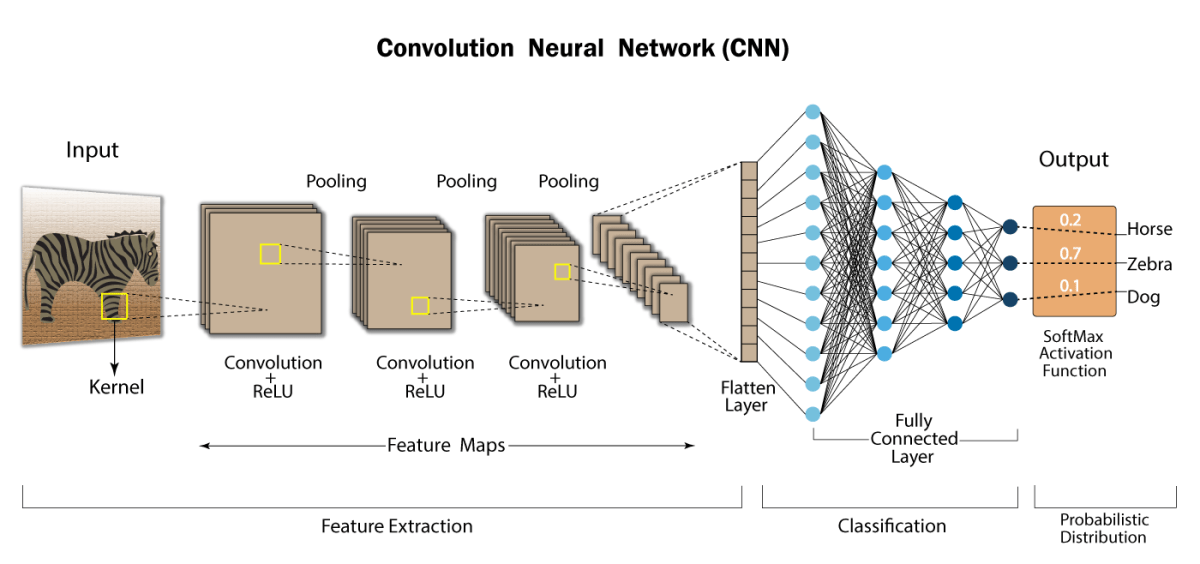


***Figure ‎2.4.2.3: pooling layer***

#### **Fully Connected Layer**

We have completed the Feature Extraction processes up to this point; the next step is Classification. The input image is classified into a label using the fully connected layer (which is what we have in ANN). This layer links the output layer to the data obtained from the preceding steps—the convolution layer and the pooling layer—and ultimately assigns the input the desired label.

The graphic below shows the full procedure for creating a CNN model.



***Figure ‎2.4.2.4: Fully connected***

## **Blender & BPY**

### **What is Blender?**

Almost all facets of 3D development are supported by Blender, a free and open-source 3D creation suite. A comprehensive 3D creation requires a solid base in modelling, as well as good texturing, rigging, animation, lighting, and a plethora of other tools. Whether you want to work exclusively with static models or venture into the world of animation, this software is fantastic.

To give you some background, the Blender Foundation, a nonprofit company founded in 2002, is where the software for Blender was created. The foundation was established in 2007 and is presently hosted by the spin-off Blender Institute, which also serves as the hub for ongoing research and development initiatives.

Despite being free, Blender is useful and accessible for a variety of users, from the novice hobbyist to the experienced animator. Even NASA makes extensive use of it for its public models! Since experienced users are constantly improving it, there may be a small learning curve for complete beginners.

At its core, Blender still revolves on access: giving individuals the tools to be creative so they can construct anything they can imagine. It's a crucial tool for individuals who want to create their own models for 3D printing.

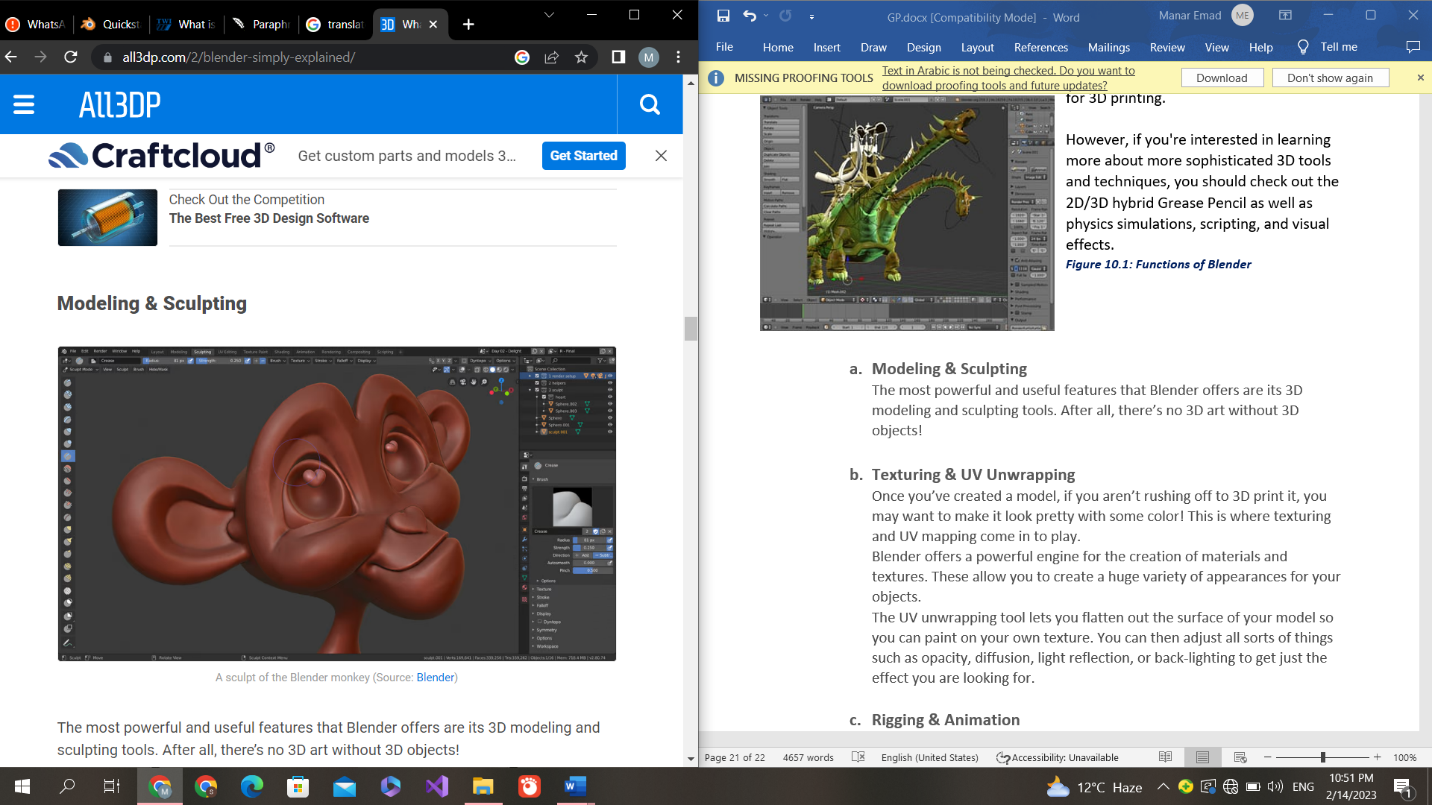
### **Features & Functions**

There are many helpful tools in Blender, however some will be more useful for beginners than others. The most used tools in Blender for new users are modelling, sculpting, and texturing, along with animation. They might not even go beyond modelling and sculpting when making products for 3D printing.

However, if you're interested in learning more about more sophisticated 3D tools and techniques, you should check out the 2D/3D hybrid Grease Pencil as well as physics simulations, scripting, and visual effects.

***Figure ‎2.5.2: Functions of Blender***

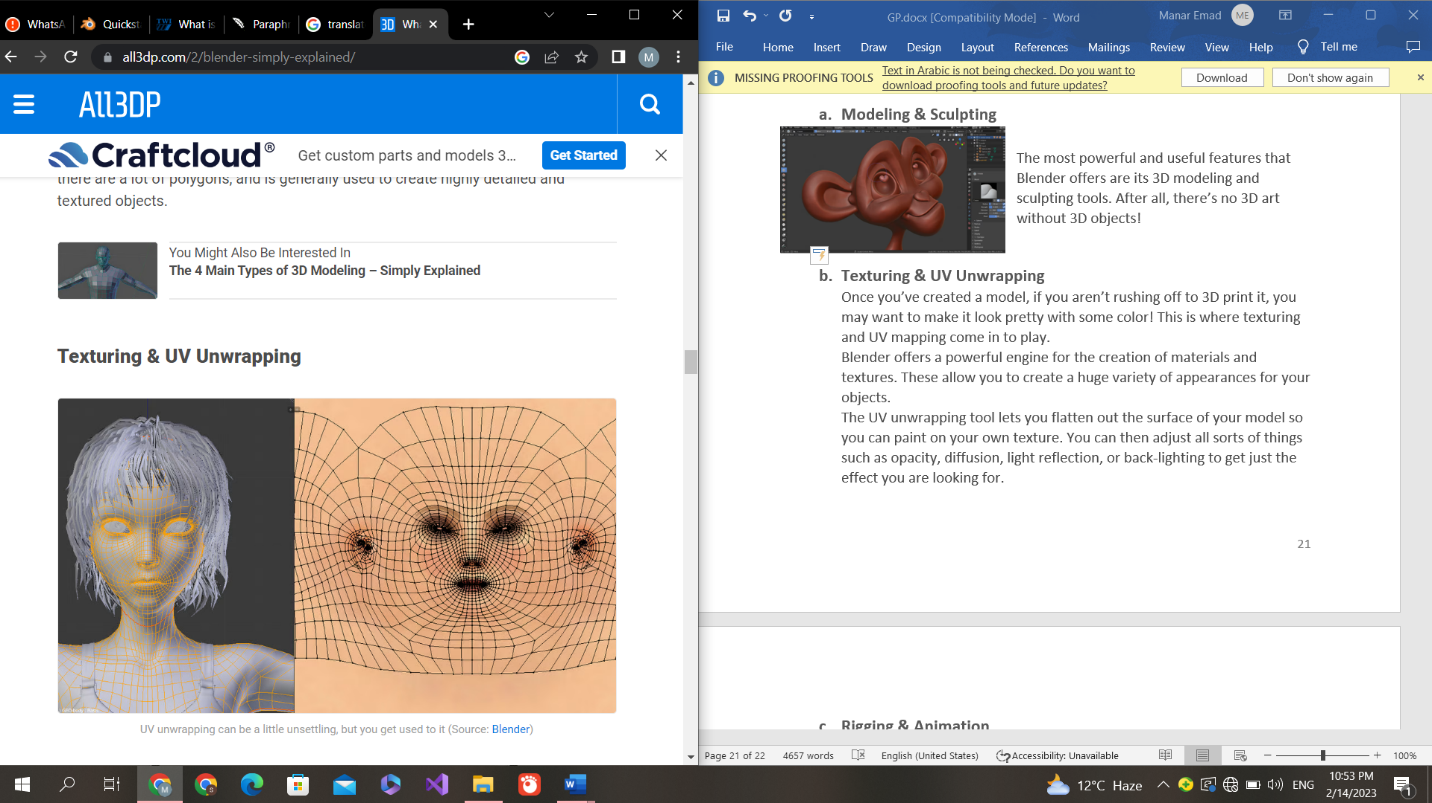
#### **Modeling & Sculpting**



The most powerful and useful features that Blender offers are its 3D modeling and sculpting tools. After all, there’s no 3D art without 3D objects!

***Figure ‎2.5.2.1******: Modeling & sculpting***

#### **Texturing & UV Unwrapping**

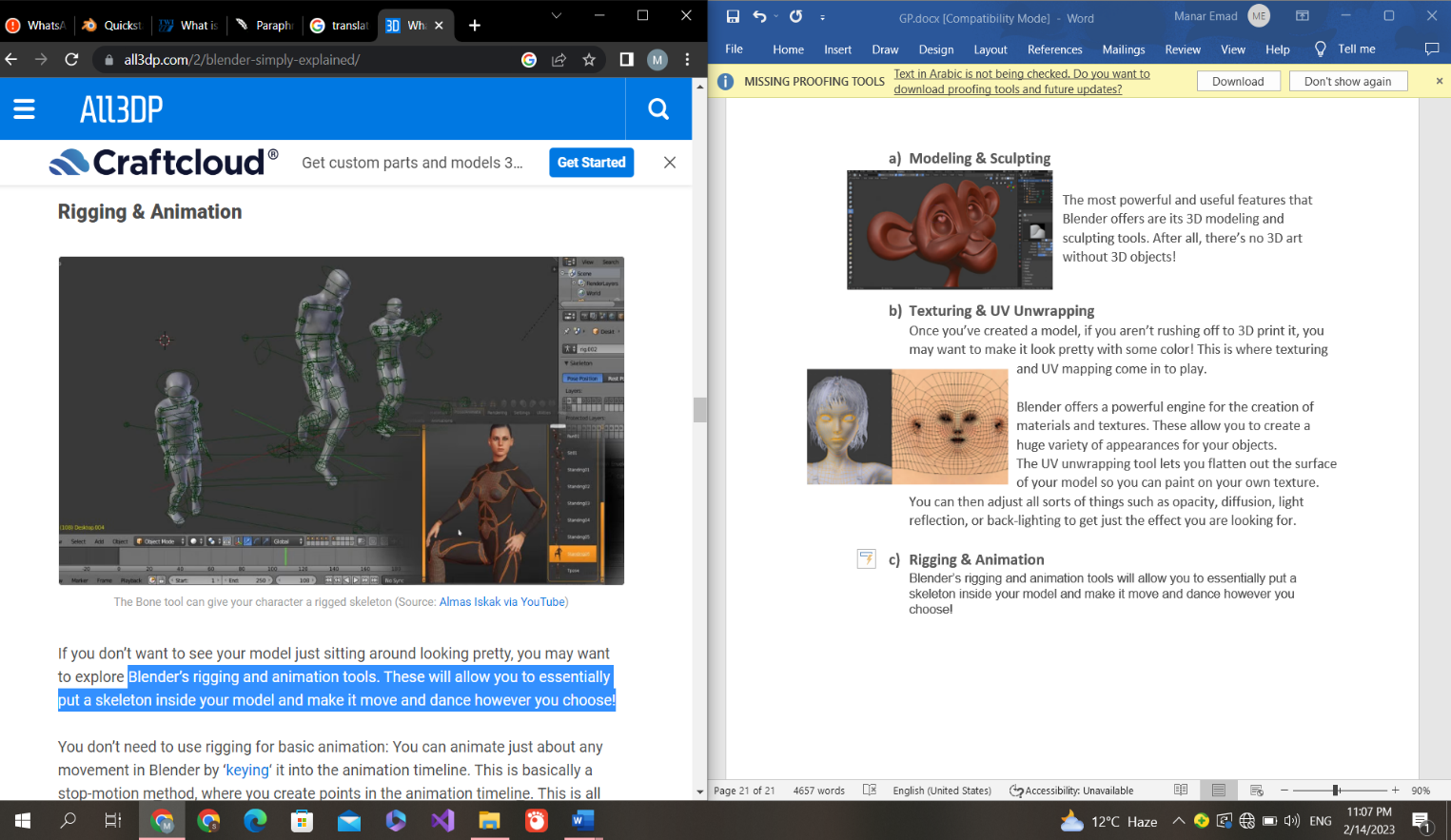
Once you’ve created a model, if you aren’t rushing off to 3D print it, you may want to make it look pretty with some color! This is where texturing and UV mapping come in to play.

***Figure ‎2.5.2.2: Texturing & UV Unwrapping***

Blender offers a powerful engine for the creation of materials and textures. These allow you to create a huge variety of appearances for your objects.

The UV unwrapping tool lets you flatten out the surface of your model so you can paint on your own texture. You can then adjust all sorts of things such as opacity, diffusion, light reflection, or back-lighting to get just the effect you are looking for.

#### **Rigging & Animation**

Blender’s rigging and animation tools will allow you to essentially put a skeleton inside your model and make it move and dance however you choose!

You don’t need to use rigging for basic animation: You can animate just about any movement in Blender by keying it into the animation timeline. This is basically a stop-motion method, where you create points in the animation timeline. This is all you need if you want something to fly around or move from point A to point B. ***Figure ‎2.5.2.3:*** ***Rigging & Animation***

For more complex animation, especially if you want to animate a character, you’ll need to use Blender’s rigging tools. With these, you can get your model moving exactly the way you want.

What’s great is that Blender will automatically fill in the movement in between your keyframes. You just need to set the starting and ending pose, and Blender will fill in the movement in between. It can take some fiddling though, which is where the art of the animator comes in.

### **BPY (Blender API)**

The bpy module is a built-in part of blender that gives python scripts access to the data and operators used in blender. This module is available to scripts run within blender itself, which includes a python interpreter.

This module can be imported in a script and gives access to Blender data, classes, and functions. Scripts that deal with Blender data will need to import this module.

This module defines properties to extend Blender's internal data. The result of these functions is used to assign properties to classes registered with Blender and can't be used directly.

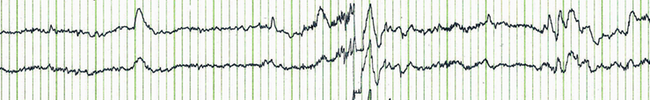
It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing. This package provides Blender as a Python module for use in studio pipelines, web services, scientific research, and more.

## **Dataset**

***Grasp-and-Lift EEG Detection***

### **Overview**

* Recall your morning routine from that time, including shutting off the alarm, getting ready, brushing your teeth, brewing coffee, enjoying a cup, and closing the door as you went for work. Now picture carrying out each of those tasks once more without the benefit of your hands.
* Patients who have lost hand function as a result of neurological impairments or amputations experience this every day. With a brain computer interface prosthetic device, a patient could regain the capacity to carry out these fundamental daily tasks, considerably enhancing their independence and quality of life. Currently, neurologically impaired people have no viable, inexpensive, or low-risk solutions for directly controlling external prosthesis with their brain activity.



***Figure ‎2.7.1: EEG Signal***

* EEG signals are evoked by brain activity and are recorded from the human scalp. Outside of particular laboratory experiments, the link between brain activity and EEG data is complicated and poorly understood. The availability of low-cost, non-invasive BCI devices that are low-risk and economical depends on improvements in the interpretation of EEG signals.
* You are required to use EEG data collected from healthy people doing the tasks of gripping, lifting, and replacing an object to determine when a hand is performing each of these actions. It is essential to comprehend the relationship between EEG signals and hand motions better in order to create a BCI device that would allow people with neurological disorders to navigate through the world more independently.

### **Dataset Description**

* This data contains EEG recordings of subjects performing grasp-and-lift (GAL) trials.
* There are 12 subjects in total, 10 series of trials for each subject, and approximately 30 trials within each series. The number of trials varies for each series. The training set contains the first 8 series for each subject. The test set contains the 9th and 10th series.
* For each GAL, you are tasked to detect 6 events:

1. HandStart

2. FurstDigitTouch

3. BothStartLoadPhase

4. LiftOff

5. Replace

6. BothReleased

*These events always occur in the same order. In the training set, there are two files for each subject + series combination:*

* the \*\_data.csv files contain the raw 32 channels EEG data (sampling rate 500Hz)
* the \*\_events.csv files contain the ground truth frame-wise labels for all events
* The events files for the test set are not provided and must be predicted. Each timeframe is given a unique id column according to the subject, series, and frame to which it belongs. The six label columns are either zero or one, depending on whether the corresponding event has occurred within ±150ms (±75frames). A perfect submission will predict a probability of one for this entire window.

# **Proposed Model**

## **Design**

***Figure 3.1: Design***

## **Preprocessing**

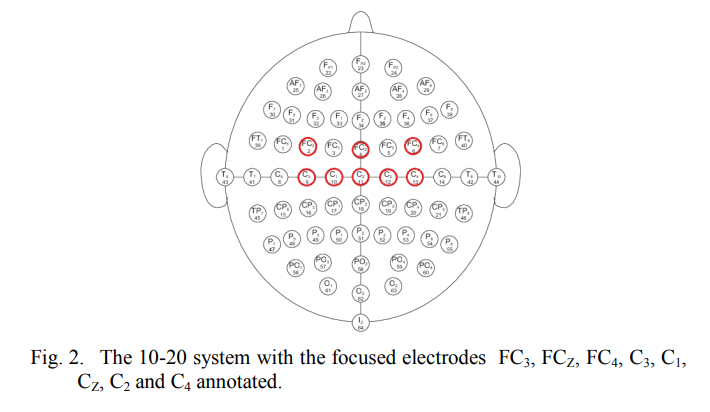
In general, preprocessing is the procedure of transforming raw data into a format that is more suitable for further analysis and interpretable for the user. In the case of EEG data, preprocessing usually refers to removing noise from the data to get closer to the true neural signals.

Preprocessing is a vital step in the analysis of EEG signals, as it can reduce noise, improve clarity, and enhance the features of the signal that are of interest. This is achieved by using various techniques, such as filtering, artifact removal, and normalization. Filtering is used to remove frequencies outside of the range of interest, such as those from muscle activity or power line interference. Artifacts can be removed through techniques such as Independent Component Analysis (ICA) or by using templates of known artifact patterns. Normalization is used to make comparisons between signals easier, by ensuring that all signals have the same scale. After preprocessing, the signal is ready for further analysis, such as feature extraction and classification. While preprocessing is a necessary step for EEG signal analysis, it is important to note that it also has the potential to introduce artifacts, so the techniques used should be carefully selected.

The first step in preprocessing EEG signals is the removal of artifacts. Artifacts are any unwanted signals that may be present in the EEG signal, such as power line noise, muscle activity, and eye movements. These artifacts can affect the accuracy of EEG data analysis and must be removed. This can be done by identifying and removing the artifact sources, or by using signal processing techniques such as Independent Component Analysis (ICA) or Temporal Independent Component Analysis (TICA).

The second step in preprocessing EEG signals is the filtering of the signal. This is done to remove any unwanted frequency components from the signal and improve the signal.

Preprocessing of inspected EEG for making easier the extraction of desired features is ponderous. Preprocessing methods used in EEG are very dependent on the goal of the applications. There are some methods that are used very commonly to improve the quality of Signal to Noise Ratio (SNR), such as Common Average Referencing (CAR). Resampling the data, filtering, bad channel detection, Independent Component Analysis (ICA), epoching continuous data, and epoch rejection are the most common techniques in the preprocessing stage of EEG recordings. Since we are interested in motor imagery tasks, we focus on manipulating data that correspond to specific electrodes over the human-patient scalp. These are recordings referred to the premotor cortex brain area. Using the typical 10- 20 system, the focused electrodes are the FC3, FCZ, FC4, C3, C1, CZ, C2, and C4



***Figure ‎3.2: The 10-20 system with focused electrodes FC3, FCZ, FC4, C3, C1, CZ, C2, and C4 annotaed.***

## **Feature Extraction**

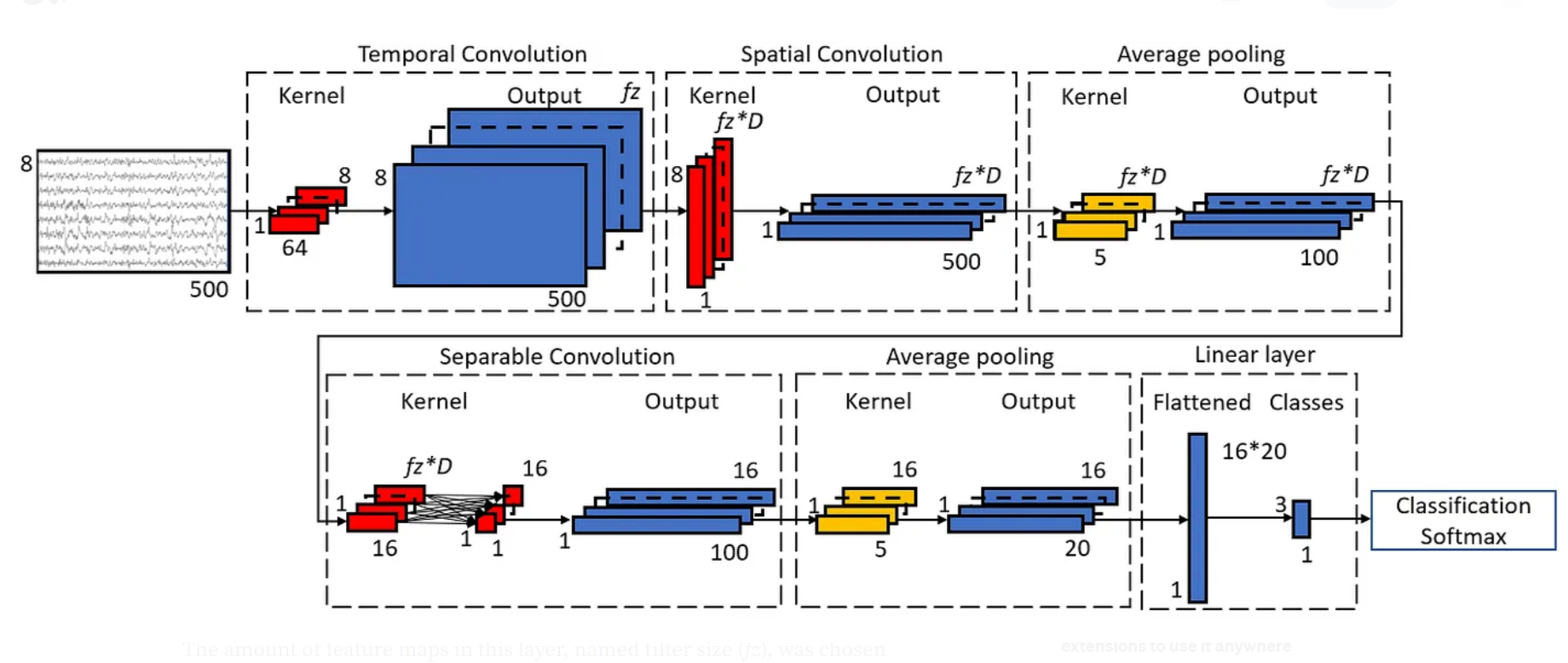
There are several reasons why preprocessing is necessary for EEG data. First, the signals picked up from the scalp are not necessarily an accurate representation of the signals originating from the brain, as the spatial information gets lost. Secondly, EEG data tends to contain a lot of noise which can obscure weaker EEG signals. Artifacts such as blinking or muscle movement can contaminate the data and distort the picture. Finally, we want to separate the relevant neural signals from random neural activity that occurs during EEG recordings. It uses matrices diagonalization of matrices to find an optimal set of spatial filters for projection, which maximizes the difference in variance values of different signals, resulting in a feature vector with a high degree of discrimination.

To simplify the feature extraction operation, Tabar et al. convert the EEG signals of each channel into time-frequency images, which are classified by a deep network stacked autoencoder (SAE).

## **EEGNET**

EEGNET consists of a temporal and spatial convolution, but also has another form of convolution, called a separable convolution.

***Figure 3.4: EEGNET Architecture***



The first layer of the network is a temporal convolution. at a size of 1 x 64. The amount of feature maps in this layer, named filter size (fz), was chosen based on hyperparameter search. Each convolution layer applied batch normalization after the convolution, to normalize the output from the previous layer to ensure a normalized input for the following layer.

The second layer is a spatial convolution of size 8 x 1. The first dimension is equal to the amount of electrode channels. The amount of feature maps from the previous layer was multiplied with a depth parameter (D), which was also chosen based on hyperparameter search. After applying batch normalization, a non-linearity was implemented with an exponential linear unit (ELU). ELU keeps output x the same when x > 0, and for x ≤ 0 the function exp(x) − 1 is applied

Then, temporal average pooling with a kernel size of 5 x 1 with a stride of 5 was applied, averaging data from each 5 timepoints to reduce dimensionality. As the input size in our study (500) was not divisible by the stride value of 8 in the original EEGNET, we chose a stride value of 5.

After average pooling, a dropout layer followed. During training, dropout randomly zeroes some of the elements of the input with a certain probability pdrop. This prevents overfitting on training data by decreasing the dependency of specific nodes to nodes in earlier layers, as having high dependency between node pairs could lead to overfitting on specific features in the training data, which would not be present in the validation and test data. The value of pdrop was found by hyperparameter search.

Next, a separable convolution layer was applied, which consisted of a temporal convolution with kernel size 1 x 16 as used in the original EEGNET, directly followed by a 1 x 1 convolution over the kernels from previous layer grouped over all the feature maps, essentially summarizing the output of the temporal convolution over the feature maps. In this layer, another batch normalization and ELU was applied.

After, another 5 x 1 average pooling was applied, followed by a dropout layer. Lastly, data was flattened, and a linear layer was applied.

All convolution layers explained above were applied with a stride of 1, without adding bias. For temporal convolutions, ‘same’ padding was used, where zeros are added to the left and right of the input to have same output size after the convolution.

As optimization method, the Adam optimizer was used. The learning rate of Adam was also found by hyperparameter search.

**Advantages**

EEGNet, a compact convolutional neural network for EEG-based BCIs that can generalize across different BCI paradigms in the presence of limited data and can produce interpretable features.

EEGNet is robust enough to learn a wide variety of interpretable features over a range of BCI tasks.

**Disadvantages**

EEGNET will most probably perform pretty bad when applying data of a single individual. Why? Simply because the amount of data is not enough for EEGNET to properly calibrate. For this reason, transfer learning (TL) has been used in the BCI field, where a model would learn from data of multiple subjects before applying the model to a new subject. However, first experiments have showed that TL for EEG data comes with a lot of difficulties and problems.

## **TCN**

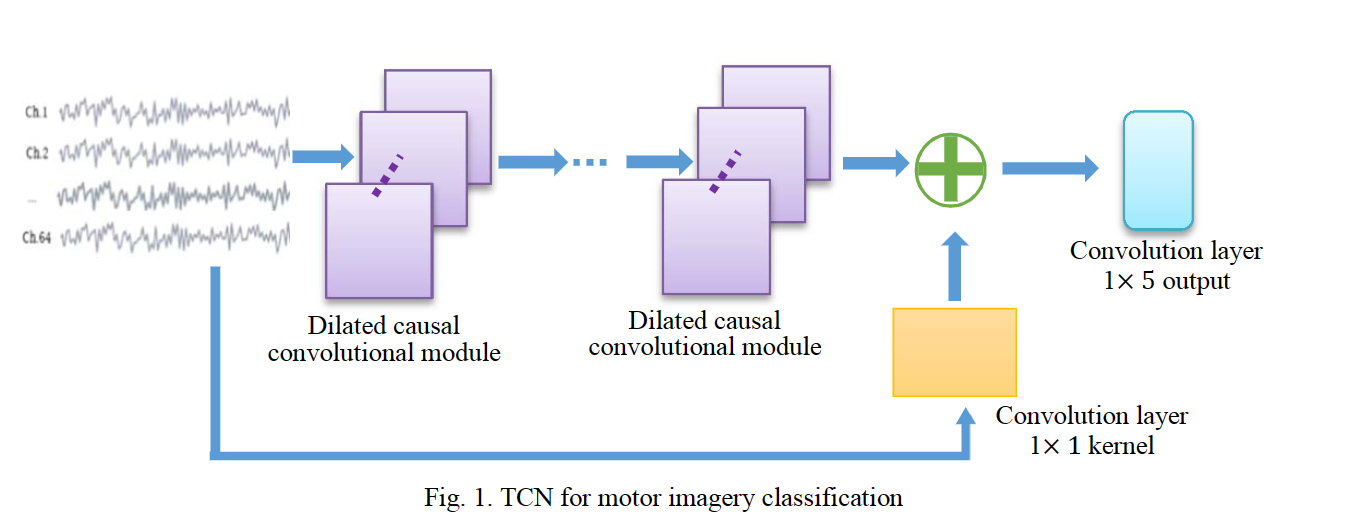
TCN could incorporate both temporal information and spatial information simultaneously through the causal convolution in it To perform causal convolution, the convolutional output at a specific time point of the corresponding convolutional layer is obtained through convolving the filter kernel with the data points at or before the output time point. To ensure the data length to be the same at the input, hidden and output layers, 1D fully convolutional layer and zero padding has been adopted in TCN.

In order to include long history information into TCN, very large convolution kernel or very deep network is required, which leads to high computational cost.

To assure a high computational efficiency and incorporate long term temporal information in the meanwhile, dilated convolution has been employed. Different from the conventional convolution, dilated convolution does not involve each entry within the data array.

The convolution is performed in a down-sampling manner. A parameter called dilation factor is used to control the dilation level. E.g., when the dilation factor is equal to 2, 1 out of 2 entries is involved in the convolution operation. When the dilation factor is 3, 1 out of 3 entries is involved in the convolution, and so on. Combined with causal convolution, dilated causal convolution is formed in TCN.

***Figure 3.5: TCN Architecture***

Within each dilated causal convolutional module, one dilated causal convolution layer is adopted, followed by a weight normalization layer, ReLU and Dropout rate of 0.05. The residual connection is built between the input and the output of the last dilated causal convolution module. A 1 × 1 kernel convolution has been added in the residual connection to adjust the length of the input and align it with the output of the last dilated causal convolution module. At the output layer, 1D fully convolution layer is adopted instead of fully connected layer

## **LSTM**

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Networks (RNNs) that are capable of learning long-term dependencies. They are particularly well suited for EEG classification tasks due to their ability to capture temporal information in the input data. LSTMs have been used to classify single-channel and multi-channel EEG signals. The architecture of an LSTM network consists of memory cells, these memory cells are connected by 'gates' which allow for the flow of information between them. This architecture allows the network to remember information over long periods of time essential for EEG classification tasks. The memory cells are connected to an input layer which takes in the EEG data and the output layer which produces the classification result. The LSTM network is trained using Backpropagation, a technique used to adjust the weights of the network to improve its performance.

The training process is Long Short-Term Memory (LSTM) networks have been gaining in popularity for classifying Electroencephalogram (EEG) data. LSTM networks are ideal for this task due to their ability to capture long-term dependencies in sequential data. EEG signals are inherently temporal, and thus, they require a model capable of considering long-term patterns and not just short-term correlations. LSTM networks have several advantages over traditional methods of EEG classification. For one, they can capture both long-term and short-term dependencies in the data. This allows them to identify patterns across the entire signal, not just within a limited time window. Furthermore, due to the nature of the LSTM network, the model can be easily adjusted to account for different EEG frequencies and temporal dynamics. Additionally, training an LSTM model for EEG classification is relatively straightforward. By providing the model with labeled EEG data, the model can quickly learn to distinguish.

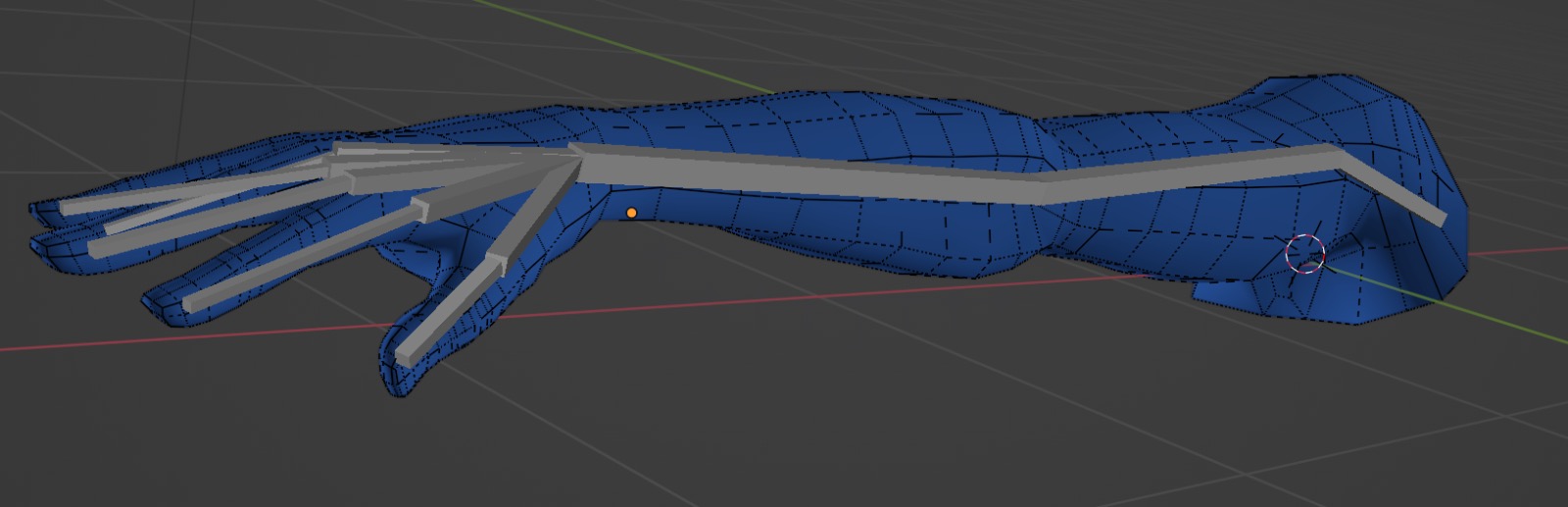
# **Results**

## **Simulation**

After the classification is done by the deep learning model, there will be an API module in which the output of the classification is taken as an input for the graphical model (a 3D prosthetic arm) built by blender application. Using the scripting environment and blender python API (BPY), we will then simulate a scene in which visualizes the output of the classification model. This will be done by running a python script which uses the blender API and contains the required functions and variables which interprets the classes of the signals and converting them to the desired inverse kinematics process. Then we will be able to render the scene that visualizes the action desired by the EEG signal. By the that, we can observe and demonstrate the digital properties of the physical prosthetic arm and enhance.

The adaptation of the arm with respect to the EEG signals in a virtual environment, and then enhance the functionality of the physical device itself.

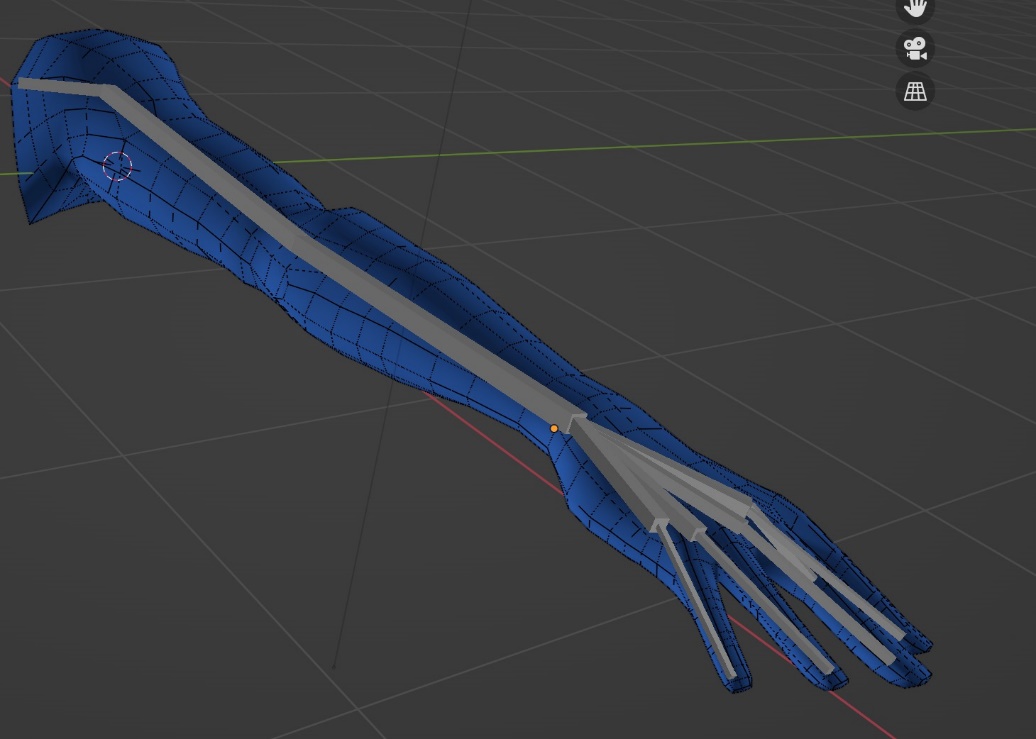
***Figure 4.1.I: ARM1***



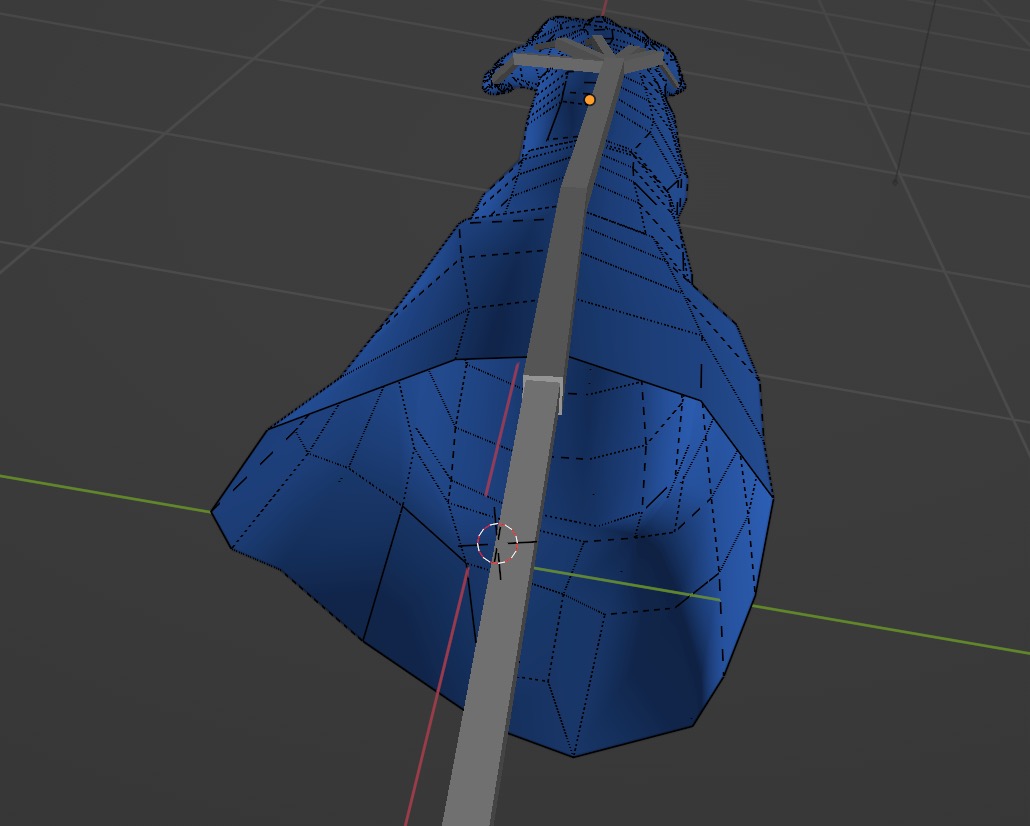
***Figure 4.1.II: ARM2***



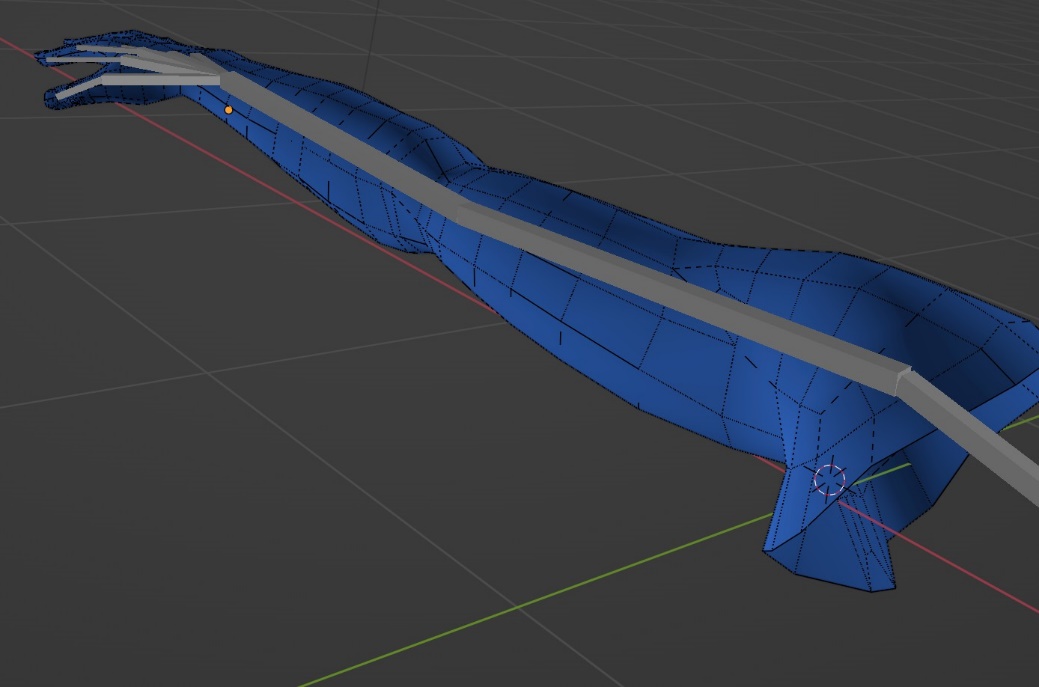
***Figure 4.1.III: ARM3***

******

***Figure 4.1.IIII: ARM4***

******

***Figure 4.1.IIIII ARM5***

******

## **Project Progress**

**Up until this point:**

* in the BCI part, the EEG Signals have been collected and preprocessed.
* The feature extraction has been finished
* As per the classification model (CNN) has been trained and tested successfully.
* In the simulation part, the 3D graphical model has been built (right hand prosthetic arm) and ready to be rendered.

**In the Coming Phase**:

* The API part will be built to integrate the BCI and simulation parts, to get the desired results of the project.

