CARPAL TUNNEL SYNDROME DETECTION USING DEEP LEARNING FOR WRONG POSE ESTIMATION ON KEYBOARD OR MOUSE

Nikita Gautam[1*] and Dr. Amit Shrivastava[2]

Article History: Received: 10.05.2022 **Revised:** 15.06.2023 **Accepted:** 20.06.2023

ABSTRACT

Carpal Tunnel Syndrome (CTS) is a kind of compressive neuropathy that may develop from excessive use of a computer's keyboard and mouse. Early identification and prevention are essential in reducing the need for surgery and potentially saving lives. This study proposes a computer management system that utilizes deep learning to detect CTS through hand gestures during virtual keyboard and mouse operations. The system detects CTS risk by monitoring keyboard and mouse poses in real-time, using a Convolutional Neural Network (CNN) machine learning (ML) model. The model's accuracy in predicting posture is high, and its performance improves with increasing epochs. Accuracy on the training data is 98.72% and on the validation data, it is 98.83% for the keyboard's posture classification model at epoch 100. Likewise, at epoch 100, the accuracy of the mouse posture classification model is 98.97% on the training data and 99.08% on the validation data.

Keywords: Carpal Tunnel Syndrome, Convolutional Neural Network (CNN), Mouse, Keyboard, Diagnosis.

[1*]Research Scholar, EEE, Poornima University, Jaipur, Plot No. IS-2027-2031, Ramchandrapura P.O. Vidhani, Sitapura, Rajasthan
nikitagtm@gmail.com , 0000-0002-3265-0636
[2]Professor, EEE, Poornima University, Jaipur, Plot No. IS-2027-2031, Ramchandrapura P.O. Vidhani, Sitapura, Rajasthan
apasjpr@gmail.com

DOI: 10.31838/ecb/2023.12.si6.593

1. INTRODUCTION

Computing technology has become essential due to significant development over the last few decades. Keyboards and mice are crucial interfaces, with optical and laser mice being the most important. The Light-emitting diode (LED) sensor-based mechanical mouse was replaced by the optical mouse, and the laser mouse further improved accuracy. The laser mouse outperforms the optical mouse on glossy surfaces [1]. Repetitive keyboard and mouse usage can result in hand stiffness, burning, and weakness, among other symptoms. These can lead to CTS, a condition caused by pressure on the median nerve due to swelling in the tunnel. Delicate finger movements and grasping small objects may also be affected [2] [3]. CTS is caused by compression of the median nerve in the wrist. It is a common type of compression neuropathy and is diagnosed through clinical complaints and nerve conduction testing. The prevalence of CTS in the general population ranges from 3.0% to 5.8% among women and from 0.6% to 2.1% among males [4]. Most people attribute carpal tunnel syndrome to pressure build-up in the tunnel at the base of the thumb. Hand-arm vibrations, frequent hand use, and the application of hand force are all biomechanical aspects that might exacerbate the illness [5].

The growing prevalence of computer usage has sparked discussions about the occupational status of CTS and the potential risk factors associated with prolonged computer use. A recent study found no significant link between computer work and CTS, but this conclusion was based on limited epidemiological research and no comprehensive literature review. Furthermore, alternative methods were not considered for managing the problem [6]. As a kind of neuropathy, Carpal Tunnel Syndrome (often abbreviated as CTS) affects 1–5% of the U.S. population [7]. Particularly susceptible to developing CTS include females, individuals over the age of 60, and desk workers [8]. CTS is caused by repetitive activities that create high pressure in the wrist. Pain and weakness in the hands are common symptoms, and surgery may be required for treatment. Early identification and prevention are essential in reducing the need for surgery and potentially saving lives [9].

1.1. Carpal Tunnel Syndrome (CTS)

Pressure on the wrist's median nerve causes carpal tunnel syndrome. The palm side of the hand has a tiny hole made of bones and ligaments. Hand and arm symptoms include weakness, numbness, and tingling [10]. A computer management system that uses hand gestures to do virtual keyboard or mouse functions is helping CTS patients. The system employs deep learning by way of a Convolutional Neural Network (CNN) to accurately predict user-input motions, therefore resolving the issues that have plagued tracking algorithms owing to the complexity of the human hand and its digits. When compared to the present norm, the proposed study gives more precise and cost-effective estimations of hand motions [11]. The suggested study provides more precise and economical hand gesture assessments than the state-of-the-art method. It became popular due to issues with misplaced computer parts and prolonged typing. There is a strong association between experiencing tingling or numbness at night and using a mouse for more than 30 hours per week [12], suggesting that mouse use might be more to blame than keyboard typing for computer workers with CTS. Carpal tunnel syndrome is shown in Figure 1.

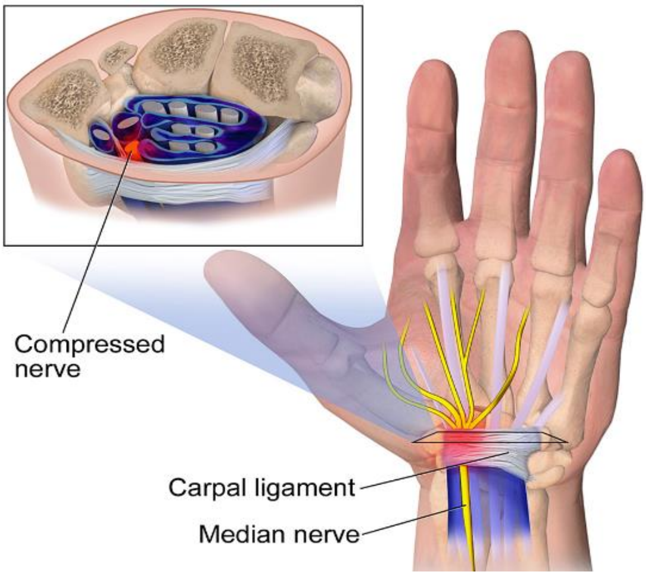


Figure 1: Carpal Tunnel Syndrome [13].

1.2. Carpal Tunnel Syndrome Diagnosis

CTS is a common condition that occurs when the median nerve in the wrist is compressed or squeezed [14]. The following are the key points for the diagnosis of Carpal Tunnel Syndrome:

- **Symptoms**

Patients with CTS usually experience symptoms such as pain, numbness, tingling, or weakness in the hand, wrist, or fingers. These symptoms are often worse at night and can affect one or both hands [15].

- **Physical Examination**

A doctor would examine the patient's hand, wrist, and arm for signs of CTS, including tenderness, swelling, and loss of sensation. They may also test the strength of the patient's grip and perform the Tinel's and Phalen's tests [16]. Medical History: Any injuries, pre-existing diseases, or drugs that might be exacerbating the patient's symptoms would be questioned by the doctor.

- **Imaging Tests**

X-rays, ultrasonography, or magnetic resonance imaging (MRI) could be recommended to confirm a diagnosis of CTS and rule out other possible causes of the patient's symptoms.

- **Nerve Conduction Study (NCS)**

Evaluation of median nerve function and CTS severity may be achieved with a nerve conduction examination.

- **Electromyography (EMG)**

Electromyography is a diagnostic test that measures the electrical activity of muscles and nerves. It can help determine the extent of nerve damage in patients with CTS [17]. The pathophysiology of CTS is shown in Figure 2:

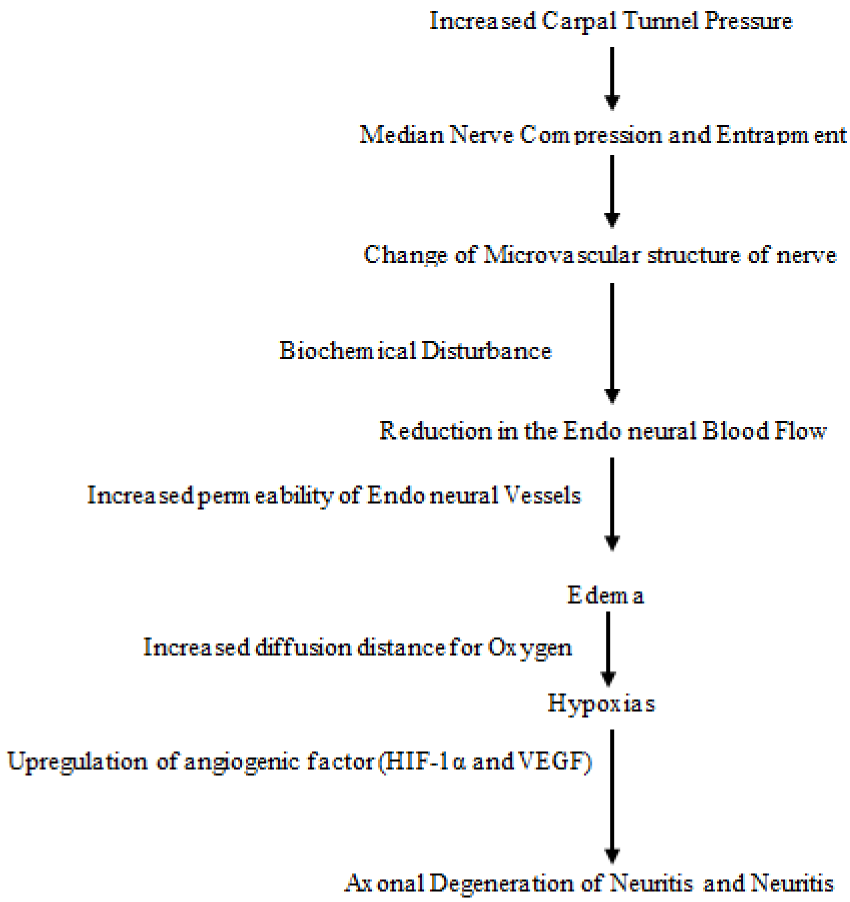


Figure 2: Carpal Tunnel Syndrome: Pathophysiology [18].

1.3. Carpal Tunnel (CT) Syndrome Symptoms and Treatment

CTS is caused by strain on the median nerve in the wrist. Early diagnosis and treatment could minimize nerve damage and symptoms.

1.3.1 Symptoms

Carpal tunnel syndrome (CTS) is a painful ailment that affects the hands and wrists. Muscle atrophy around the thumb's joint is a tell-tale sign [19]. Typical CTS symptoms include the following:

- **Numbness or Tingling**

The most prominent sign of CTS is a lack of sensation in the fingers. The thumb, other three digits, and ring finger are often affected, but the pinky finger is spared.

- **Pain**

Pain is another common symptom of CTS, which may be felt in the wrist, hand, or forearm.

- **Weakness**

CTS can also cause weakness in the hand, making it difficult to hold onto objects or grip them tightly.

- **Burning Or Itching Sensation**

Some people with CTS may experience a burning or itching sensation in the affected hand or fingers.

- **Loss of Coordination**

Buttoning garments or typing on a keyboard are only two examples of fine motor abilities that might be affected by CTS.

- **Worsening Symptoms at Night**

Many people with CTS experience more severe symptoms at night, which can disrupt sleep.

1.3.2 Treatment

The treatment options available for CTS include surgical release of the carpal tunnel, nonsteroidal anti-inflammatory drugs (NSAIDs), medication injections, rehabilitation techniques such as stretching, ASTM Advantage, strengthening, and therapeutic ultrasound, as well as immobilization with splinting.

- **NSAIDs**

Common dosing for ibuprofen is 800 mg twice a day for 7-10 days [20]. Anti-inflammatory drugs and steroid pills are typical treatments [21].

- **Injection**

Relief from corticosteroid injections often lasts between two and four months, but only 22 percent of patients report no symptoms 18 months following treatment [22]. Conservative therapy with steroid injections was beneficial for those with mild to severe CTS [23].

- **Splinting**

Neutral nocturnal wrist support and a 15-degree cock-up splint are commonly used. Research has not shown significant evidence that splinting reduces intracarpal pressure [24]. Patients who received splinting and NSAIDs experienced more discomfort at follow-up than those who underwent a basic range of motion training) [25].

- **Rehabilitation Modalities**

Pulsed ultrasound treatments for CTS are as effective as drug injections and wrist splinting [26]. Manipulative therapy can safely and effectively treat mild to moderate CTS [27]. ASTM AdvantEDGE is a manual therapy approach that uses hand-held equipment to administer microtrauma to connective tissue structures [28], promoting fibrosis absorption and soft tissue regeneration [29] [30]. Stretching and strengthening exercises can improve wrist extension and avoid the need for surgery [31][32].

- **Surgery**

Carpal tunnel decompression surgery is the only permanent cure for CTS [33], but not necessarily in all cases [34]. Around 40% of CTS cases are treated with surgery, costing over $1.9 billion annually [35]. Open-release surgery between tendons is typically performed with an early range of motion advocated to reduce the possibility of irritations forming on the median nerve [36]. Surgery provides good to great pain relief for 86% of patients, but only 40% recover to normal functioning, and 5% experience increasing pain [37]. CTS surgeries are common, but there is no agreement on postoperative care or recovery time [38].

This study offers a potentially useful strategy for managing CTS through advanced deep learning methods, which could have a significant impact on society by improving the quality of life and productivity of those affected by this condition.

2. LITERATURE OF REVIEW

This section provides a summary of the studies done on the topic of utilizing deep learning to diagnose Carpal Tunnel Syndrome.

Di Cosmo et al. (2022) [39] studied that Ultrasound imaging could be used to detect CTS because of the excellent visualization of the median nerve that it provides. However, ultrasonography is not routinely used to assess entrapment neuropathy due to limitations such as the condition's sensitivity to the skill of the evaluator and the absence of standardized methods. The study suggests that sonographers be given a helping hand in isolating the median nerve using ultrasound images by using a completely automated deep-learning method. While the proposed method was able to correctly detect and segment the median nerve in pictures with normal morphology, it struggled with images with unusual anatomical details. The data collection that supports sonographers in their daily job would grow to cover additional facets of normal anatomy and disease as more is understood.

Smerilli et al. (2022) [40] introduced a convolutional neural network (CNN)-based deep learning system to automatically analyze the median Nerve's CSA in ultrasound images taken at the carpal tunnel's proximal entrance. Dice similarity coefficient, precision, recall, mean average precision, and recall were utilized to evaluate testing data outcomes. After reducing anatomical variations, the algorithm detected the median nerve in 41 of 49 (83.7%) and 41 of 43 (95.3%) photos. Deep learning has improved the ultrasonography evaluation of carpal tunnel syndrome.

Zhou et al. (2022) [41] explained that CTS is one of the most common types of peripheral nerve sickness in adults, and it could end a person's life and career. Patients are identified and treated more slowly due to the use of universal diagnostic criteria. Deep CTS uses deep learning to analyze MR images for signs of carpal tunnel syndrome. In experiments, the deep learning system outperformed networks trained on micro-objects. They used 333 images for training and 82 for testing, achieving an accuracy of 0.63 for intersection over union and a segmentation efficiency of 0.17 seconds, both of which are encouraging for use in clinical settings.

A. W. H, Ng. et al. (2021) [42] analyzed the significance of MRI in diagnosing and evaluating CTS, which is the most common type of peripheral nerve entrapment. It also emphasizes that MRI is useful in detecting subclinical or unusual CTS symptoms, and it can aid in ruling out secondary causes of the condition. The study discusses both common and uncommon symptoms of CTS before and after surgery and notes that radiologists may be the first to identify atypical symptoms. The paper concludes by stressing the importance of MRI in managing and diagnosing CTS.

Tsamis et al. (2021) [43] examined the possibility of using machine learning and electrodiagnostic features to detect median nerve mononeuropathy. The results indicate that automated electrodiagnosis can be reliable and accurate, with a 95% accuracy rate compared to the gold standard neurophysiological diagnosis and 89% accuracy rate compared to clinical diagnosis. This method can be useful in diagnosing carpal tunnel syndrome, as it provides increased precision with minimal human influence. Combining unique traits with standard methods can further enhance accuracy.

Horng et al. (2020) [44] studied that CTS is a condition commonly found in individuals who use strong or vibrating manual tools in their jobs. Ultrasound imaging is used for diagnosis and treatment of this condition. Most image analysis methods, including the active contour model, need manual median nerve segmentation. However, the Deep Nerve model uses Mask Track features with convolutional long-short-term memory to automatically recognize and segment the median nerve. Accuracy (0.8912), recall (0.9119), and F-score (0.9015) were assessed for the model. The student's t-test demonstrated significant variations in median nerve size, circumference, aspect ratio, and circularity.

Wei et al. (2020) [45] presented an ML method that uses Random Forests (RF) to analyze grip strength data and identify variables that best predict Carpal Tunnel Syndrome (CTS). Although CTS is acknowledged as a work-related ailment, there are currently no established recommendations for early screening. The method isolates defining features of functional control of the hand that is prognostically associated with stages that indicate the incidence and severity of CTS. The study's results suggest the usefulness of this approach in identifying CTS among high-risk groups.

Deepa et al. (2020) [46] evaluated the problem of CTS brought on by extended computer use by developing a method of computer administration that uses hand gestures to conduct computer keyboard or mouse operations. The system employs deep learning algorithms, specifically a CNN approach, to estimate user-input movements accurately and quickly in gesture recognition. The study overcame the challenge of tracking hand movements due to hand anatomy and quick finger movements by using the CNN approach. The proposed approach is accurate, fast, and effective in estimating hand gestures to prevent CTS.

3. BACKGROUND STUDY

The pressure of the median nerve in the carpal tunnel causes CTS. To provide suitable treatment, it's crucial to determine CTS severity. Although Machine Learning (ML) and Deep Learning (DL) have advanced medical research, there is limited work on CTS and no ML model based on extensive clinical data for identifying CTS severity. In this study, ML algorithms were used to develop CTS severity classification models. 80 CTS patients and 80 patients with similar illnesses like cervical radiculopathy, de Quervain tendinopathy, and peripheral neuropathy underwent US-guided median nerve hydro dissection. CTS was classified into minimal, moderate, or severe categories. 70% of the dataset was used for training and 30% for testing. The suggested model has a classification accuracy of 0.955%, a precision of 0.963%, and a recall of 0.919%. The author created a machine learning model that predicts a patient's likelihood of improvement following hydro-dissection injection (at months 1, 3, and 6). The model's accuracy was 0.912% after six months, 0.901% after three months, and 0.877% after one month. Predictions after six months exceed those after one and three months. The author used statistics (significance testing, Spearman's correlation testing, and a two-way Analysis of Variance (ANOVA) test) to examine the injection process's influence on CTS therapy. The data-driven decision-support tools may assist patients in selecting their surgeries to reduce the risks and costs associated with surgery [47].

4. PROBLEM FORMULATION

Carpal tunnel syndrome has a significant economic burden on society, affecting the quality of life, productivity, and function, as well as incurring high treatment costs. It is more common in women and worsens with age, with risk factors including obesity, pregnancy, and various health conditions. However, there are limitations in clinical diagnosis, as tests may not detect smaller fibers responsible for pain. A proposed solution is a computer management system utilizing deep learning, employing a Convolutional Neural Network to detect CTS through hand gestures during virtual keyboard and mouse operations. The method includes pre-processing and extracting significant characteristics from data to distinguish healthy and unhealthy stances and diagnose CTS in the user. This cutting-edge approach offers a potentially useful strategy for managing CTS through advanced deep learning methods.

5. RESEARCH OBJECTIVES

- To develop a deep learning-based algorithm that can accurately detect Carpal Tunnel Syndrome by analyzing keyboard or mouse usage patterns.
- To investigate the relationship between incorrect hand and wrist posture during keyboard and mouse usage and the development of carpal tunnel syndrome.
- To design and implement a system that can provide real-time feedback to users about their keyboard or mouse poses, and alert them if they are at risk of developing Carpal Tunnel Syndrome.

- To diagnose the stage of Carpal Tunnel Syndrome by analyzing other acquired data, indicating the severity of the syndrome.

6. RESEARCH METHODOLOGY

Techniques for research are discussed about the idea of planned architecture.

6.1 Dataset Generation

The following methods are crucial in the generation of datasets:

6.1.1 Pin Configuration

For constructing the carpal tunnel syndrome detection device VCC +5V pin of Adruino UNO is connected to VCC pin of MPU6050 and to the 10k Ohm resister which was connected to the broader side of flex sensor. The Arduino ground pin is linked to the MPU6050 ground pin and the flex sensor's thinner side. The MPU6050's SCL pin is wired to Arduino's analog pin A5. The MPU6050's SDA pin is wired into position A4. The broader side of 3 flex sensor is connected to analog pins A0, A1 and A2 of Arduino. The pin configuration is shown in figure 3.

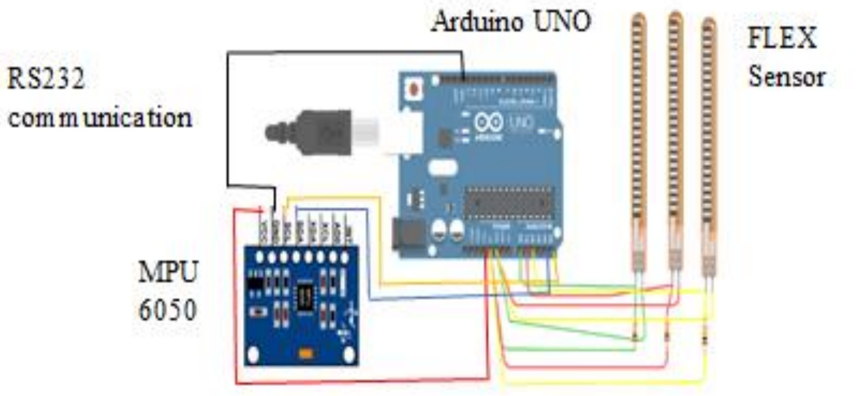
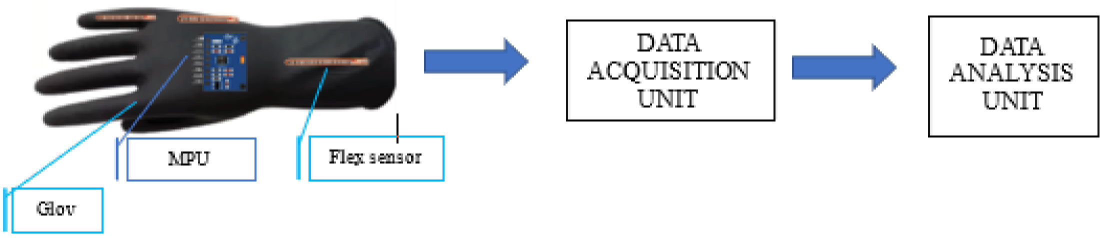


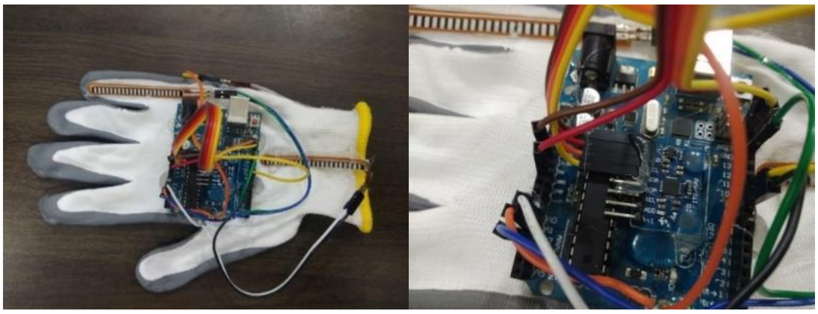
Figure 3: Pin Configuration of Hardware.

6.1.2 Proposed System

Figure 4 shows a block schematic of the proposed system. Two auxiliary procedures, a data collection unit, and an analysis unit, are used to implement the suggested method.



(a)



(b)

Figure 4: (a) System Model, and (b) Hands-on hardware architecture.

6.1.3 Data Acquisition Unit

This unit reads data from the carpal tunnel syndrome detection device and sends it to the data analysis unit for further processing. The fabricated carpal tunnel syndrome detection device consists of MPU6050 which is a six-axis accelerometer, 3 FLEX sensor which is a type of variable resistance which vary with the bend and an Arduino UNO microcontroller. The process starts with the parameter settings of MPU 6050. The data held in the cache is cleared. DPS processing signal is set to 2000. Data labels, range, clock source and other related settings are done appropriately. MPU 6050 reads acceleration at all three-axis and sends it to Arduino UNO's pins A4 and A5 by I2C protocol and the Flex sensor reads the bend at multiple positions and sends it to Arduino UNO's pins A0, A1 and A2 by I2C protocol. The data received by Arduino UNO is in raw form therefore data normalization and separation take place. After data normalization, the data is stored. Figure 5 depicts the data acquisition unit's process flow diagram.

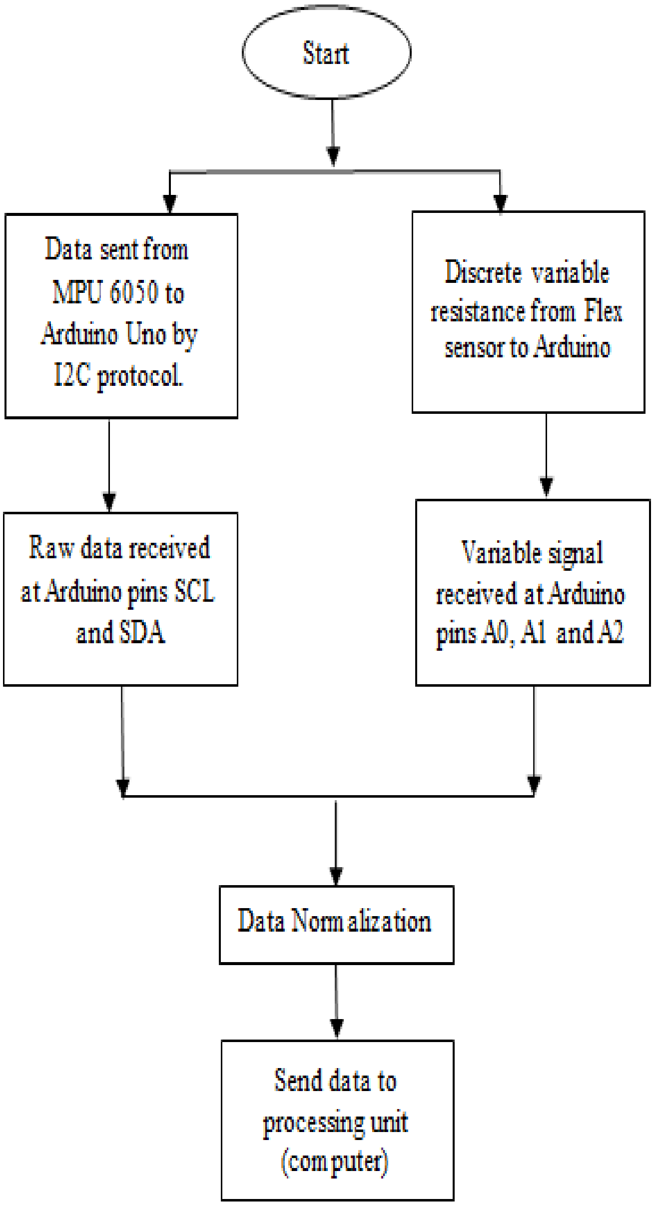


Figure 5: Process Flow for Data Acquisition from Hardware

6.1.4 Data Analysis Unit:

This unit aims to continuously receive data from the data acquisition unit and detect movement and bend of the palm. This unit can be classified into two subunits namely the testing unit and the training unit. The training unit consists of datasets of the vibration signatures obtained while regular activities such as working with a mouse and different vibration signatures obtained during different movements of the palm. Send data to processing unit Data Normalization Data sent from MPU 6050 and 3 FLEX sensors to Arduino Uno by I2C protocol Raw data received at Arduino pins A0, A1, A2, A4, and A5: Process Flow for Data Acquisition active in real-time. The task of the training unit is to acquire data from data acquisition and identify the movement and bend in the palm by contrasting the data with the dataset. The process flow diagram for the data acquisition unit is shown in Fig 6.

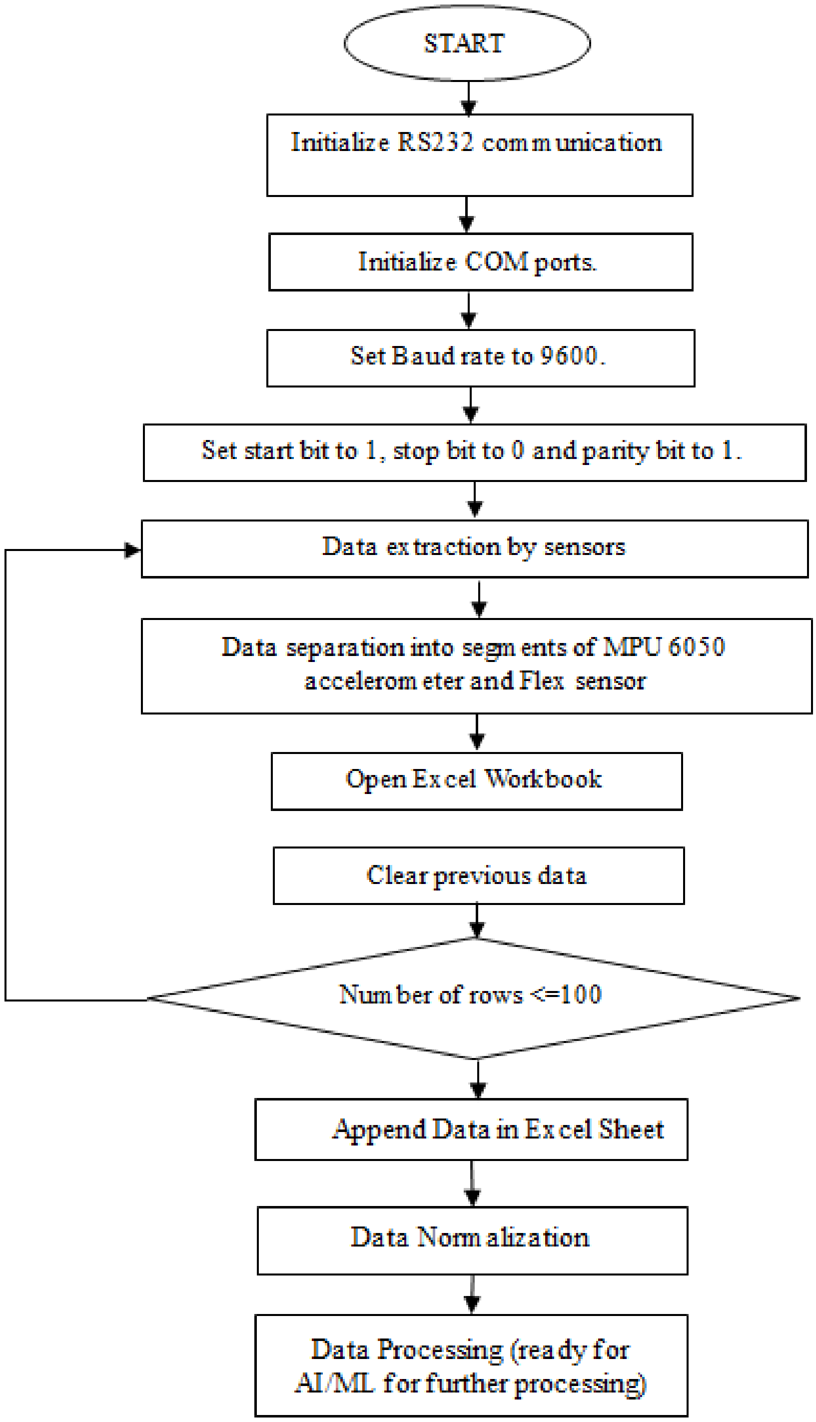


Figure 6: Process Flow for Data Analysis (AI/ML) Unit

6.2 Technique Used

6.2.1 Convolutional Neural Network (CNN)

CNN is also known as ConvNet, and it is a category of Artificial Neural Networks (ANN) with a deep feed-forward construction and incredible simplifying capabilities when associated with more networks with Fully Connected (FC) layers [48].

This could grasp highly conceptual elements of things, especially spatial data, and identify them efficiently. A deep CNN architecture consists of a constrained number of computational layers, each of which could learn many tiers of abstraction from the image it is fed as input. The initiatory stages are responsible for learning and extracting the higher-level traits, whereas the deeper layers are responsible for learning and extracting the lower-level characteristics [48]. Figure 7. Depicts CNN's core conceptual paradigm.

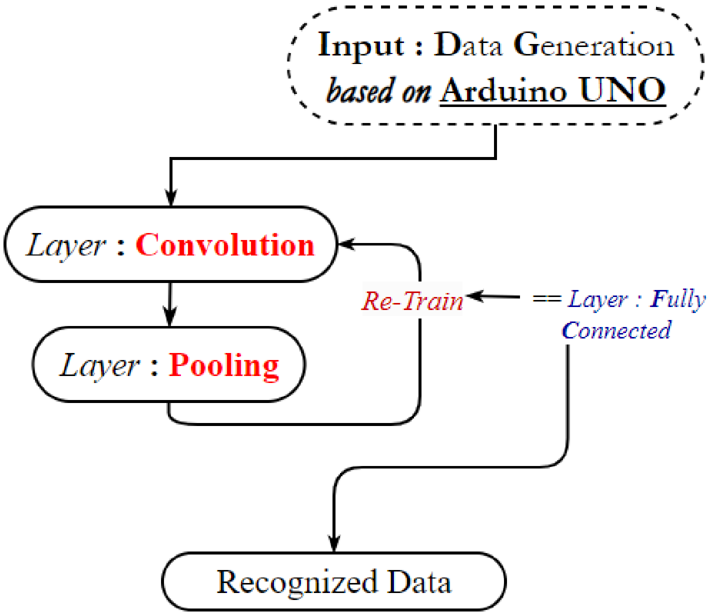


Figure 7: CNN [48].

6.2.2 Deep Belief Networks (DBNs)

Deep Belief Networks (DBNs) are artificial neural networks used for text feature extraction [49]. DBNs include numerous layers of hidden units and are learned using unsupervised learning techniques like Contrastive Divergence (CD) or Restricted Boltzmann Machines (RBMs). In text data, DBNs can be used to extract useful features from the raw text by learning a hierarchical representation of the data. Each layer of the DBN learns to represent the data at a different level of abstraction, starting from simple features such as individual words and building up to more complex features such as syntactic and semantic relationships between words.

DBNs can be trained on large amounts of unlabeled text data, which is an advantage in scenarios where labeled data is limited. Once the DBN is trained, the learned features can be used as input to a classifier or other downstream tasks, such as sentiment analysis or topic modeling. DBNs have shown promising results in feature extraction for text data, and they are a powerful tool for natural language processing tasks. However, they can be computationally expensive to train and require careful tuning of hyperparameters, which can make them challenging to use in practice. The basic formula for DBNs is as follows:

$$h = f(Wx + b) \qquad (1)$$

where:

H is the hidden layer, which represents the learned features.

F is the activation function, which introduces non-linearity into the model.

W is the weight matrix, which is learned during training.

X is the input data, which in the case of text data, could be a bag-of-words or TF-IDF representation.

b is the bias term, which is also learned during training.

6.3 Proposed Algorithm

Input: Let **X** be the **input dataset** containing mouse and keyboard usage data, including correct and incorrect postures.

Step 1: Data Pre-processing
Let **Z** be the **pre-processed data** after applying text pre-processing techniques, cleaning, normalization, and noise removal.

Step 2: Feature Extraction
Let **F** be the **extracted features** using ***Deep Belief Networks (DBN)*** or other suitable feature extraction techniques.

Step 3: Split the dataset.
Partition the data into **training set (X_train, Y_train)** and **testing set (X_test, Y_test).**

Step 4: Train and Test the Model
Let **M** be the **machine learning model**, such as a ***Convolutional Neural Network (CNN),*** trained using the training set (X_train, Y_train). Then, the trained model is evaluated using the testing set (X_test, Y_test).

Step 5: CTS Detection and Identification
Apply the **trained model (M)** to the pre-processed data (Z) to **classify** correct and incorrect **postures.** Use the classification results to **detect and identify Carpal Tunnel Syndrome** in the user.

Step 6: CTS Stage Diagnosis
Analyze other acquired data, such as the duration of mouse and keyboard usage, to **diagnose the stage of CTS** and its severity.

Output: Let **Y** be the **output,** which is the diagnosis of Carpal Tunnel Syndrome (CTS) and its stage.

Therefore, the mathematical formula for the given algorithm can be represented as:

Y = CTS_Diagnosis_and_Stage(X)

Where **CTS_Diagnosis_and_Stage** is a **function** that performs the following steps:

- Z = Data_Preprocessing (X)
- F = Feature_Extraction(Z)
- (X_train, Y_train), (X_test, Y_test) = Split_Dataset (Z, Y)
- M = Train_Model (F, X_train, Y_train)
- performance = Test_Model (M, X_test, Y_test)
- CTS_Classification = Apply_Model (M, Z)
- CTS_Diagnosis_and_Stage = Diagnose_CTS (CTS_Classification, X, other_data)

6.4 Proposed Methodology
The steps taken to complete the requested task are outlined below. The suggested method's block diagram could be shown in Figure 8. Step-by-step instructions for the proposed method's process would be provided.

Step 1: Obtain the Dataset as Input Data.
The first step is to obtain a dataset that contains data related to mouse and keyboard usage, including correct and incorrect postures while using them.

Step 2: Data Pre-Processing
In this step, the dataset undergoes data pre-processing operations such as data cleaning, normalization, and text pre-processing techniques like stemming or lemmatization. These operations help to remove noise and inconsistencies from the data, making it more suitable for feature extraction and modeling.

Step 3: Feature Extraction
Following cleaning, the data is subjected to feature extraction methods such as Deep Belief Networks (DBN). These techniques help to extract relevant features from the data that are useful for training a model.

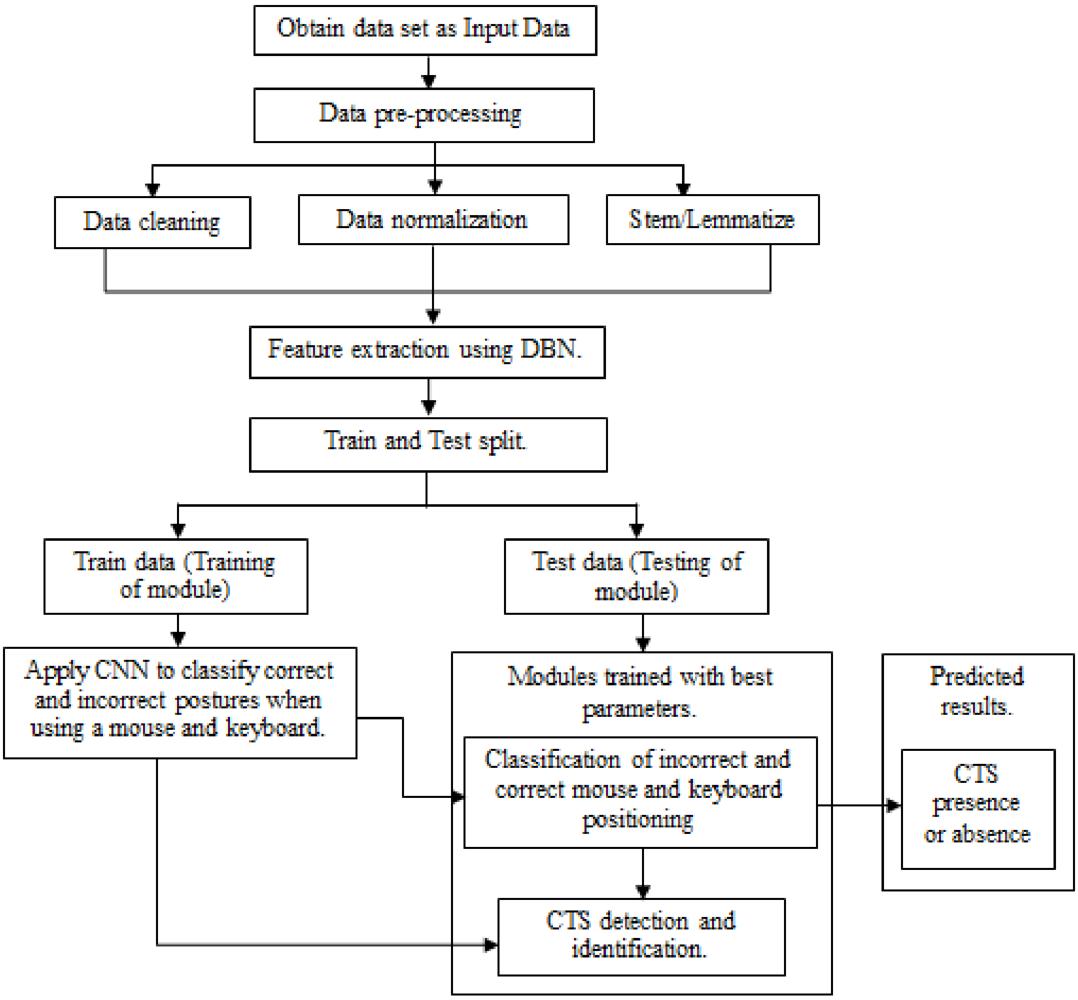


Figure 8: Proposed Methodology

Step 4: Split Data into Train and Test Sets
The information is divided into a training set and an evaluation set. The model is "trained" on the training set, and its efficacy is "tested" on the testing set.

Step 5: Train and Test the Model.
In this step, the model is first trained on the train data and then validated on the test data. This process is useful for testing the model's efficacy.

Step 6: Apply CNN to Classify Correct and Incorrect Postures.
CNN is a type of neural network that is used for classification. In this step, the CNN model is applied to the data to classify correct and incorrect postures when using a mouse and keyboard.

Step 7: CTS Detection and Identification
This step involves using the results obtained from the previous steps to detect and identify Carpal Tunnel Syndrome in the user.

Step 8: Diagnosis
Finally, the approach can be used to diagnose the stage of CTS by analyzing other acquired data, indicating the severity of the syndrome.

7. RESULT AND DISCUSSION

A keyboard and mouse posture prediction machine learning model's performance is analyzed in the outcome section. The training and validation dataset's loss and accuracy values across 100 epochs are provided. The model performed best in the latter epochs, with the greatest accuracy and lowest loss values. The data are summarized in tables, and graphical representations are provided to help visualize the trends in the data. Overall, the results indicate that the model was able to predict posture with high accuracy, and the authors can use the data to make informed decisions about the effectiveness of the model and identify any potential issues that need to be addressed. The discussion section may elaborate on the implications of the results and potential future directions for research in this area.

7.1 Keyboard Posture Analysis

The data provided represents the performance of a machine learning model during its training process. Model training lasted 100 epochs, during which time loss and accuracy measurements were taken on both the training and validation sets. Table 1 has five columns:

- Epochs represents the number of passes the model has made through the entire training dataset during the training process. The table has five columns:
- Loss is a value that represents how well the model can predict the target output compared to the actual output. A low loss value indicates a better model performance.
- Accuracy represents how often the model predicted the correct output compared to the actual output. A higher accuracy value indicates a better model performance.
- Val_loss is representing the loss value for the validation dataset. This value is used to evaluate the model performance on data that it has not seen during training. A low val_loss indicates a better model performance.
- Val_accuracy represents the accuracy value for the validation dataset. This value is used to evaluate the model performance on data that it has not seen during training. A higher val_accuracy indicates a better model performance.

Epoch 1 had a high loss value of 1.7342, low accuracy of 44.64% and a high val_loss of 0.9813 with a low val_accuracy of 70.13%. This indicates that the model had a poor performance during the initial training epoch. However, as the number of epochs increased, the loss values decreased, and the accuracy values increased for both training and validation datasets.

The highest accuracy values were achieved during the later epochs, with epoch 100 having the highest accuracy value of 98.72%. Similarly, the lowest loss values were achieved during the later epochs, with epoch 100 having the lowest loss value of 0.1792. The validation dataset also showed a similar pattern, with the highest validation accuracy of 98.83% and the lowest validation loss of 0.172 beings achieved during epoch 100. Table 1 below summarizes the data for each epoch.

Table 1: Data for each epoch.

Epochs	Loss	Accuracy	val_loss	val_accuracy
1	1.7342	44.64	0.9813	70.13
10	0.8017	72.55	0.8047	68.72
20	0.593	78.91	0.5591	85.13
30	0.4573	84	0.4766	90.04
40	0.3658	88.09	0.4028	93.41
50	0.3351	89.45	0.3054	92.99
60	0.2905	90.82	0.2818	95.51
70	0.2513	92.36	0.2412	96.01
80	0.2256	93.82	0.1774	96.07
90	0.1841	95.36	0.1748	97.34
100	0.1792	98.72	0.172	98.83

In Figure 9, the Training Loss and Validation Loss are compared graphically based on 100 epochs of data. The model's performance during training is shown in this graph. By analyzing Training Loss and Validation Loss patterns, the author could evaluate the model's efficacy and detect any flaws. Figure 9 helps assess the model's performance and ensure its optimum operation.

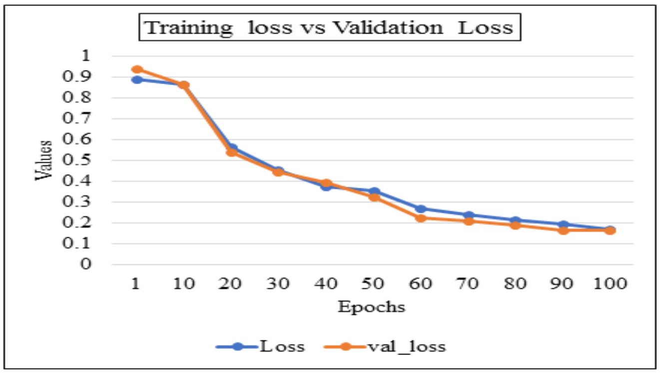


Figure 9: Comparison between Validation Loss and Training loss.

Figure 10 shows a 100-epoch comparison between Training and Validation Accuracy. During training and validation, the graph shows model accuracy changes. By analyzing the graph, the author can gain insights into the performance of the model and make necessary adjustments to improve its accuracy.

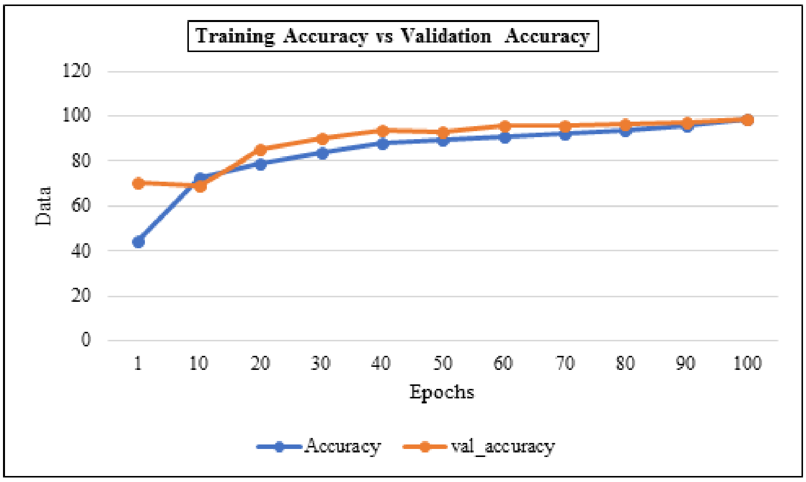


Figure 10: Comparison between Training Accuracy and Validation Accuracy.

7.2 Mouse Posture Analysis

Table 2 below summarizes the data on mouse posture, including the loss and accuracy values for each epoch, as well as the validation loss and validation accuracy. For each epoch, the loss and accuracy (percentage of correctly classified samples) are reported for both the training and validation sets. The table shows that as the number of epochs increases, the loss values decrease, and accuracy values increase for both the training and validation sets. This indicates that the model is learning to predict the correct posture for the mouse with increasing accuracy as more training iterations are performed. The highest validation accuracy achieved was 99.08% at the final epoch, indicating that the model is highly accurate in predicting mouse posture.

Table 2: Loss and Accuracy Values for Each Epoch.

Epochs	Loss	Accuracy	Val_loss	Val_accuracy
1	0.8922	46.32	0.9413	72.61
10	0.8677	79.83	0.8647	71.43
20	0.5630	81.62	0.5391	85.24
30	0.4563	83.22	0.4460	90.15
40	0.3758	88.46	0.3928	92.82
50	0.3551	90.051	0.3254	93.43
60	0.2705	91.31	0.2218	95.73
70	0.2413	92.764	0.2112	96.54
80	0.2156	93.11	0.1874	97.21
90	0.1941	94.726	0.1648	97.42
100	0.1692	98.97	0.1652	99.08

Figure 11 depicts a graphical representation of the disparity between the training and validation loss for mouse posture. The visualization showcases the variance in the loss values for both the training and validation datasets, highlighting the possible overfitting or underfitting of the model. By examining the trend lines of the two curves, one can identify the optimal point of the model, where the loss is minimized, and the performance is optimized.

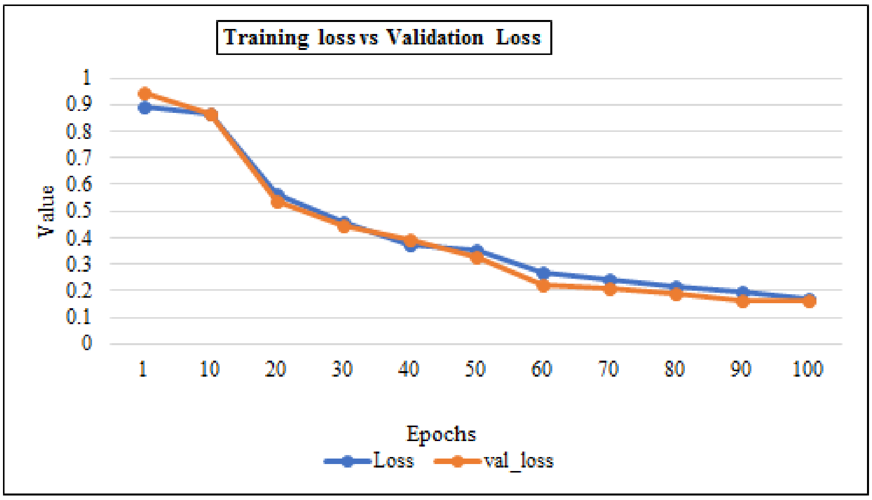


Figure 11: Loss values

A graph, shown in Figure 12, displays the difference between training and validation accuracy over 100 epochs for a mouse posture dataset. The graph plots the training accuracy and validation accuracy on the y-axis against the number of epochs on the x-axis. The data for the graph is presented in a table, which lists the accuracy and validation accuracy for each epoch. The results show that the training accuracy steadily improves over the 100 epochs, starting at 46.32% in the first epoch and reaching 98.97% by the 100th epoch. The validation accuracy also improves overall, but not as consistently as the training accuracy. The validation accuracy starts at 72.61% in the first epoch and reaches a peak of 99.08% by the 100th epoch, but it fluctuates more in the intermediate epochs.

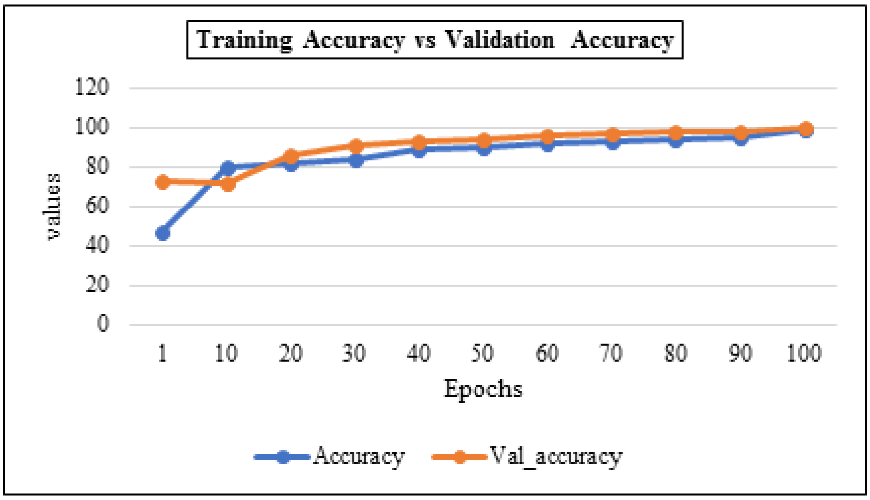


Figure 12: Accuracy Values.

7.3 DISCUSSION

The data provided includes information on the posture of both the keyboard and mouse and the performance of machine learning models during their training processes. For keyboard posture, the model had a poor performance during the initial training epoch, but as the number of epochs increased, the loss values decreased, and the accuracy values increased for both training and validation datasets. The highest accuracy values were achieved during the later epochs. Similarly, for mouse posture, the model learned to predict the correct posture with increasing accuracy as more training iterations were performed. At the last epoch, validation accuracy reached 98.83%. The model's performance during training could be shown in graphical representations like Figure 7 and Figure 9 for keyboard and mouse posture, respectively. To ensure the model is neither overfitting nor underfitting the data, accuracy and loss measurements must be monitored.

CONCLUSION

Computing technology has become an essential part of daily life, and keyboards and mice are crucial interfaces. However, repetitive keyboard and mouse usage can result in Carpal Tunnel Syndrome (CTS), which can cause symptoms such as hand stiffness, burning, and weakness. CTS is a common type of compression neuropathy and is diagnosed through clinical complaints and nerve conduction testing. Although there is some disagreement on whether CTS constitutes an occupational ailment and whether computer usage is a risk factor, it is clear that early detection and prevention are crucial for minimizing the need for surgery and perhaps saving lives. The proposed solution to manage CTS involves the use of a computer management system utilizing deep learning to detect CTS through hand gestures during virtual keyboard and mouse operations. The data on performance of the keyboard posture classification model for each epoch is measured using Loss and Accuracy metrics on both training and validation data. At the first epoch, the model had a high loss of 1.7342 and a low accuracy of 44.64%. However, as the number of epochs rises, the accuracy of the model greatly improves. At epoch 100, the model achieves a very low loss of 0.1792 and an impressive accuracy of 98.72% on the training data. The validation data also shows a similar trend of improvement in model performance with increasing epochs. At epoch 1, the model has a very high validation loss of 0.9813 and an accuracy of 70.13%. However, at epoch 100, the model achieves an impressively low validation loss of 0.172 and a high accuracy of 98.83%. A mouse posture classification model was trained for 100 epochs and its performance was evaluated based on loss and accuracy values. The model exhibited a training loss of 0.8922 an accuracy of 46.32% and a validation loss of 0.9413 and an accuracy of 72.61% at epoch 1. The model's performance improved as the training continued, with decreasing loss and increasing accuracy values. At epoch 100, the model had a training loss of 0.1692 and an accuracy of 98.97%. The validation loss was 0.1652, with a validation accuracy of 99.08%. The system can provide real-time feedback to users about their keyboard or mouse poses, alert them if they are at risk of developing CTS and diagnose the stage of CTS by analyzing other acquired data. The results of the machine learning model's performance show that it was able to predict posture with high accuracy, indicating its potential effectiveness in managing CTS. Future research could investigate how the system could be implemented in real-world scenarios to detect and manage CTS effectively.

ACKNOWLEDGEMENT

The author would like to express her sincere gratitude to " Prof. Dr. Amit Shrivastava" from the "Poornima University, Jaipur." For their invaluable contributions to this research paper. Their expertise and insights have been instrumental in shaping the direction of this study. The author is also grateful for the support and resources provided by the "Poornima University, Jaipur “. The author extends his heartfelt thanks to all those who have contributed to the successful completion of this research.

REFERENCES

[1] Pereira, C. A. M., Neto, R. B., Reynaldo, A. C., de Miranda Luzo, M. C., & Oliveira, R. P. (2009). Development and evaluation of a head-controlled human-computer interface with mouse-like functions for physically disabled users. Clinics, 64(10), 975-981.

[2] Deepa, R., and M. K. Sandhya. "An efficient hand gesture recognition system using deep learning." In Intelligent Computing, Information and Control Systems: ICICCS 2019, pp. 514-521. Springer International Publishing, 2020.

[3] Thomsen, J.F., Gerr, F., Atroshi, I.: Carpal tunnel syndrome and the use of computer mouse and keyboard: a systematic review. BMC Musculoskelet Disord. 9, 134 (2008). https://doi. org/10.1186/1471-2474-9-134

[4] Palmer KT, Harris EC, Coggon D: Carpal tunnel syndrome and its relation to occupation: a systematic literature review. Occup Med (Lond) 2007, 57:57-66.

[5] Atroshi I, Gummesson C, Ornstein E, Johnsson R, Ranstam J: Carpal tunnel syndrome and keyboard use at work: a populationbased study. Arthritis Rheum 2007, 56:3620-3625.

[6] Thomsen, Jane F., Fred Gerr, and Isam Atroshi. "Carpal tunnel syndrome and the use of computer mouse and keyboard: a systematic review." BMC musculoskeletal disorders 9, no. 1 (2008): 1-9.

[7] P. J. Ostergaard, M. A. Meyer, and B. E. Earp, "Non-operative treatment of carpal tunnel syndrome," Current reviews in musculoskeletal medicine, vol. 13, no. 2, pp. 141–147, 2020

[8] Carpal tunnel syndrome. Mayo Clinic. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/ carpal- tunnelsyndrome/symptoms-causes/syc-20355603

[9] Ayrapetyan, Mariam, Wenny Miao, and Kellen O'Rourke. "Over-Use Warning Glove for Carpal Tunnel Syndrome."

[10] https://www.google.com/search?q=Carpal+tunnel+syndrome+researchgate&source=lmns&bih=913&biw=1280&rlz=1C1RXQR_enIN1021IN1021&hl=en-US&sa=X&ved=2ahUKEwjnuM285Mb-AhUXwqACHU_ lC Z AQ_AUoAHoECAEQAA

[11] Deepa, R., and M. K. Sandhya. "An efficient hand gesture recognition system using deep learning." In Intelligent Computing, Information and Control Systems: ICICCS 2019, pp. 514-521. Springer International Publishing, 2020.

[12] Akinbinu, Tope R., and Y. J. Mashalla. "Impact of computer technology on health: Computer Vision Syndrome (CVS)." Medical Practice and Reviews 5, no. 3 (2014): 20-30.

[13] Joshi, Aditya, Karan Patel, Aleem Mohamed, Solomon Oak, Michelle H. Zhang, Hailey Hsiung, Alex Zhang, and Urvish K. Patel. "Carpal Tunnel Syndrome: Pathophysiology and Comprehensive Guidelines for Clinical Evaluation and Treatment." Cureus 14, no. 7 (2022).

[14] Duncan SFM and Kakinoki R. Carpal tunnel syndrome and related median neuropathies: Challenges and complications. Carpal Tunn. Syndr. Relat. Median Neuropathies Challenges Complicat; 2017. 1–315. DOI: 10.1007/978-3-319-57010-5.

[15] De Krom MCTFM. et al. Risk factors for carpal tunnel syndrome. American Journal of Epidemiology; 1990. 132(6):1102–1110.

[16] Wipperman J and Goerl K. Carpal tunnel syndrome: diagnosis and management. Am. Fam. Physician; 2016. 94(12):993–999. DOI: 10.5435/okoj-12-11-3.

[17] Annisa, Devi, Sri Budhi Rianawati, Masruroh Rahayu, Neila Raisa, and Shahdevi Nandar Kurniawan. "Carpal tunnel syndrome (diagnosis and management)." Journal of Pain, Headache and Vertigo 2, no. 1 (2021): 5-7.

[18] Duncan SFM and Kakinoki R. Carpal tunnel syndrome and related median neuropathies: Challenges and complications. Carpal Tunn. Syndr. Relat. Median Neuropathies Challenges Complicat; 2017. 1–315. DOI: 10.1007/978-3-319-57010-5.

[19] https://my.clevelandclinic.org/health/diseases/4005-carpal-tunnel-syndrome

[20] Chang MH, Chiang HT, Lee SSJ, Ger LP, Lo YK. Oral drug of choice in carpal tunnel syndrome. Neurology 1998; 51: 390 ± 393.

[21] Feuerstein M, Burrell LM, Miller VI, Lincoln A, Huang GD, Berger R. Clinical management of carpal tunnel syndrome: a 12- year review of outcomes. American Journal of Industrial Medicine 1999; 35: 232 ± 245

[22] Branco K, Naeser MA. Carpal tunnel syndrome: clinical outcome after low-level laser acupuncture, microamps transcutaneous electrical nerve stimulation, and other alternative therapiesÐan open protocol study. Journal of Alternative Complementary Medicine 1999; 5(1): 5 ± 26.

[23] Dammers JW, Veering MM, Vermeulen M. Injection with methylprednisolone proximal to the carpal tunnel: Randomized double-blind trial. British Medical Journal 1999; 319(7214): 884 ± 886

[24] Luchetti R, Schoenhuber R, Alfarano M, Deluca S, De Cicco G, Landi A. Serial overnight recordings of intracarpal canal pressure in carpal tunnel syndrome patients with and without wrist splinting. Journal of Hand Surgery (Br) 1994; 19: 35 ± 37

[25] Banta CA. A prospective, nonrandomized study of iontophoresis, wrist splinting, and anti-in¯ammatory medication in the treatment of early-mild carpal tunnel syndrome. Journal of Occupational Medicine 1994; 36: 166 ± 168.

[26] Oztas O, Turan B, Bora I, Karakaya MK. Ultrasound therapy e€ ect in carpal tunnel syndrome. Archives of Physical Medicine and Rehabilitation 1998; 79: 1540 ± 1544

[27] Ebenbichler GR, Resch KL, Nicolakis P, Wiesinger GF, Uhl R, Ghanem AH, Fialka V. Ultrasound treatment for treating the carpal tunnel syndrome: Randomized `sham' controlled trial. British Medical Journal 1998; 316: 731 ± 735

[28] Sucher BM, Hinrichs RN. Manipulative treatment of carpal tunnel syndrome: Biomechanical and osteopathic intervention to increase the length of the transverse carpal ligament. Journal of the American Osteopathic Association 1998; 98(12): 679 ± 686.

[29] Sucher BM. Palpatory diagnosis and manipulative management of carpal tunnel syndrome. Journal of the American Osteopathic Association 1994; 94: 647 ± 663

[30] Baker D, Wilson JK. Bilateral carpal tunnel syndrome in a piano teacher. Physical Therapy Case Reports 1999; 2(2): 73 ± 76.

[31] Groves EJ, Rider BA. A comparison of treatment approaches used after carpal tunnel release surgery. American Journal of Occupational Therapy 1989; 43: 398 ± 402.

[32] Cook AC, Szabo RM, Birkholz SW, King EF. Early mobilization following carpal tunnel release: A prospective randomized study. Journal of Hand Surgery (Br) 1995; 20: 228 ± 230.

[33] Kerwin G, Williams CS, Seiler JG. The pathophysiology of carpal tunnel syndrome. Hand Clinics 1996; 12(2): 243 ± 251.

[34] Levine DW, Simmons BP, Koris MJ, Daltroy LH, Hohl GG, Fossel AH, et al. A self-administered questionnaire for the assessment of severity of symptoms and functional status in carpal tunnel syndrome. Journal of Bone and Joint Surgery (American) 1993; 75A: 1585 ± 1592.

[35] Hirooka T, Hashizume H, Senda M, Nagoshi M, Inoue H, Nagashimja H. Adequacy and long-term prognosis of endoscopic carpal tunnel release. Acta Medicine Okayama 1999; 53(1): 39 ± 44.

[36] Cook AC, Szabo RM, Birkholz SW, King EF. Early mobilization following carpal tunnel release: A prospective randomized study. Journal of Hand Surgery (Br) 1995; 20: 228 ± 230.

[37] Armstrong MB, Villalobos RE. Surgical treatment of carpal tunnel syndrome. In: GH Kraft, EW Johnson (eds) Physical Medicine and Rehabilitation Clinics of North America, Carpal Tunnel Syndrome. Philadelphia: W.B. Saunders, 1997; 8(3): 529 ± 539.

[38] Wilson, J. K., and T. L. Sevier. "A review of treatment for carpal tunnel syndrome." Disability and rehabilitation 25, no. 3 (2003): 113-119.

[39] Di Cosmo, Mariachiara, Maria Chiara Fiorentino, Francesca Pia Villani, Emanuele Frontoni, Gianluca Smerilli, Emilio Filippucci, and Sara Moccia. "A deep learning approach to median nerve evaluation in ultrasound images of carpal tunnel inlet." Medical & Biological Engineering & Computing 60, no. 11 (2022): 3255-3264.

[40] Smerilli, Gianluca, Edoardo Cipolletta, Gianmarco Sartini, Erica Moscioni, Mariachiara Di Cosmo, Maria Chiara Fiorentino, Sara Moccia, Emanuele Frontoni, Walter Grassi, and Emilio Filippucci. "Development of a convolutional neural network for the identification and the measurement of the median nerve on ultrasound images acquired at carpal tunnel level." Arthritis Research & Therapy 24, no. 1 (2022): 38.

[41] Zhou, Haiying, Qi Bai, Xianliang Hu, Ahmad Alhaskawi, Yanzhao Dong, Zewei Wang, Binjie Qi et al. "Deep CTS: a Deep Neural Network for Identification MRI of Carpal Tunnel Syndrome." Journal of Digital Imaging (2022): 1-12.

[42] Ng, A. W. H., J. F. Griffith, and I. S. H. Ng. "MRI of carpal tunnel syndrome: before and after carpal tunnel release." Clinical Radiology 76, no. 12 (2021): 940-e29.

[43] Tsamis, Konstantinos I., Prokopis Kontogiannis, Ioannis Gourgiotis, Stefanos Ntabos, Ioannis Sarmas, and George Manis. "Automatic electrodiagnosis of carpal tunnel syndrome using machine learning. "Bioengineering 8, no. 11 (2021): 181.

[44] Horng, Ming-Huwi, Cheng-Wei Yang, Yung-Nien Sun, and Tai-Hua Yang. "DeepNerve: a new convolutional neural network for the localization and segmentation of the median nerve in ultrasound image sequences." Ultrasound in Medicine & Biology 46, no. 9 (2020): 2439-2452.

[45] Wei, Yuan, Wei Zhang, and Feng Gu. "Towards diagnosis of carpal tunnel syndrome using machine learning." In 2020 3rd Artificial Intelligence and Cloud Computing Conference, pp. 76-82. 2020.

[46] Deepa, R., and M. K. Sandhya. "An efficient hand gesture recognition system using deep learning." In Intelligent Computing, Information and Control Systems: ICICCS 2019, pp. 514-521. Springer International Publishing, 2020.

[47] Elseddik, Marwa, Reham R. Mostafa, Ahmed Elashry, Nora El-Rashidy, Shaker El-Sappagh, Shimaa Elgamal, Ahmed Aboelfetouh, and Hazem El-Bakry. "Predicting CTS Diagnosis and Prognosis Based on Machine Learning Techniques." Diagnostics 13, no. 3 (2023): 492.

[48] Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. In Recent Trends and Advances in Artificial Intelligence and Internet of Things (pp. 519-567). Springer, Cham

[49] Lopes, Noel, Bernardete Ribeiro, Noel Lopes, and Bernardete Ribeiro. "Deep belief networks (DBNs)." Machine Learning for Adaptive Many-Core Machines-A Practical Approach (2015): 155-186.