

T.C.  
**MİLLÎ EĞİTİM BAKANLIĞI**  
**Mesleki ve Teknik Eğitim Genel Müdürlüğü**

**16. ULUSLARARASI MEB ROBOT YARIŞMASI SERBEST PROJE KATEGORİSİ**  
**PROJE RAPORU**

**ROBOT NUMARASI ve ADI : 41839 / GİPROV (Görüntü İşleme ve  
Yapay Sinir Ağları ile Protez Verimliliği ve Sorun Tespit)**

**TEMATİK ALAN : ENGELSİZ YAŞAM TEKNOLOJİLERİ**

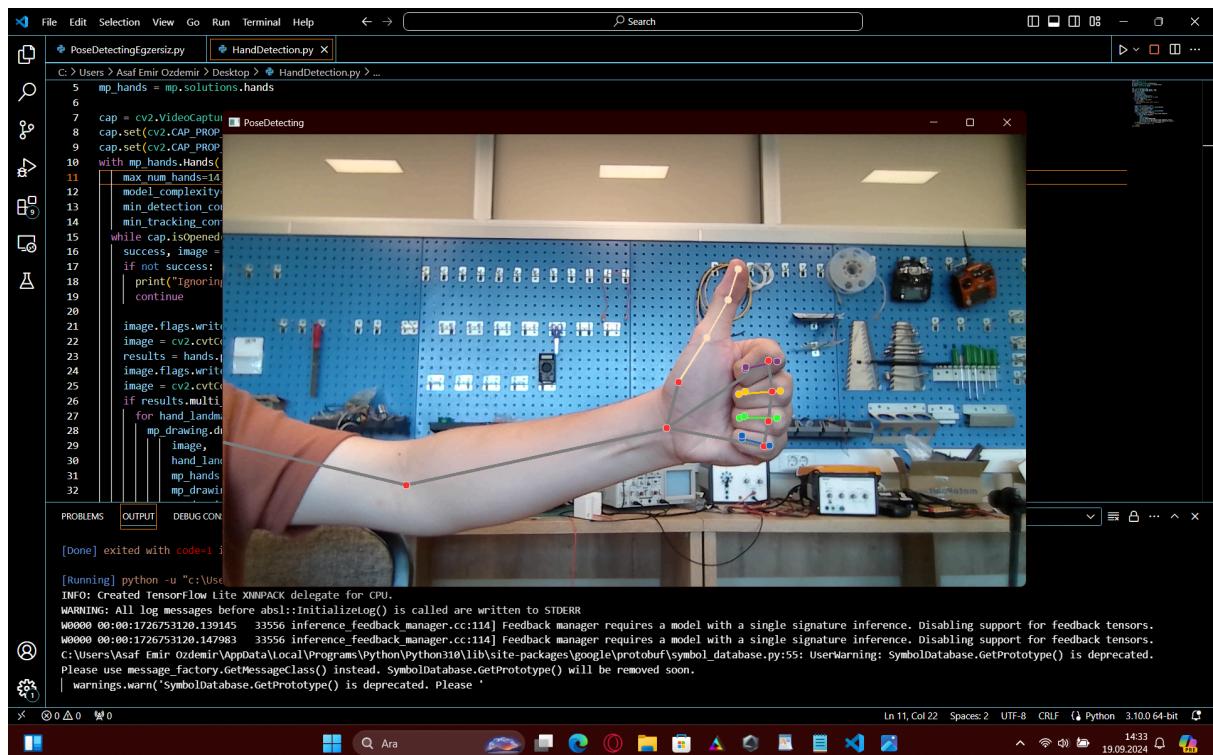
**ÜNİVERSİTE / LİSE ADI : İSTANBUL ATATÜRK FEN LİSESİ**  
**ÖĞRENCİ ADI VE SOYADI : ASAF EMİR ÖZDEMİR**  
**DANIŞMAN ADI ve SOYADI : METİN TOPRAK**

# İÇİNDEKİLER

|   |           |
|---|-----------|
| <b>Özet.....</b>  | <b>4</b>  |
| <b>1. Amaç.....</b>   | <b>5</b>  |
| <b>1.1. Problem.....</b>  | <b>5</b>  |
| 1.1.1. Zaman İçerisinde Meydana Gelen Problemler.....           | 5         |
| 1.1.2. Başlangıç Temelli Problemler.....                        | 5         |
| <b>1.2. Projenin Amacı.....</b>                                 | <b>6</b>  |
| <b>1.3. Hedef Kitle.....</b>                                    | <b>6</b>  |
| <b>1.4. Ekonomik Veriler Ve Verimlilik.....</b>                 | <b>6</b>  |
| <br>  |           |
| <b>2.1. Yenilik Unsurları Ve Literatür Taraması.....</b>        | <b>6</b>  |
| <b>2.1. Literatür Taraması.....</b>                             | <b>6</b>  |
| <b>2.2. Literatür Özeti.....</b>                                | <b>7</b>  |
| <b>2.3. Yenilik Unsurları Ve Literatürdeki Boşluklar.....</b>   | <b>7</b>  |
| <br>  |           |
| <b>3. Yöntem.....</b>   | <b>8</b>  |
| <b>3.1. Görüntü İşleme.....</b>                                 | <b>8</b>  |
| <b>3.2. Yapay Sinir Ağısı.....</b>                              | <b>8</b>  |
| 3.2.1. Algoritma.....   | 8         |
| 3.2.1.1. Yapay Sinir Ağının Temel Değişkenleri.....             | 8         |
| 3.2.1.2. Sigmoid Aktivasyon Fonksiyonu.....                     | 9         |
| 3.2.1.3. Yapay Sinir Ağrı Zincir Kuralı.....                    | 9         |
| 3.2.1.4. Hidden Unit'lerin Değişim Değerinin Bulunması.....     | 9         |
| 3.2.1.5. Ağırlıkların (Weight) Değişim Değerinin Bulunması..... | 10        |
| 3.2.1.6. Çıktı Verme.....                                       | 11        |
| 3.2.1.7. Algoritma Sonucu.....                                  | 11        |
| 3.3. Kullanılan Uygulamalar Ve Yazılımlar.....                  | 11        |
| <b>3.4. Veri Seti.....</b>                                      | <b>12</b> |
| <b>3.5. Algoritma Akış Şeması.....</b>                          | <b>12</b> |
| <br>  |           |
| <b>4. İş, Süreç Ve Planlama.....</b>                            | <b>12</b> |
| <br>  |           |
| <b>5. Sonuç.....</b>  | <b>13</b> |
| <b>5.1. Bulgular.....</b>                                       | <b>13</b> |
| <b>5.2. Analizler.....</b>                                      | <b>13</b> |
| <b>5.3. Sonuç Ve Tartışma.....</b>                              | <b>14</b> |
| <br>  |           |
| <b>6. Geliştirilebilirlik.....</b>                              | <b>14</b> |
| <b>7. Öneriler.....</b>   | <b>14</b> |
| <b>Kaynakça.....</b>  | <b>15</b> |

## ÖZET

Ampute, vücuttaki bir uzvun (kol, bacak, el vb.) belirli bir bölgesinin veya tamamının cerrahi olarak alınmasıdır. Hastada bulunan bi enfeksiyon, yaralanma veya dolaşım bozukluğu; ampute işleminin gerçekleşmesine sebep olabilir. Ampute kişiler, bir uzvunu kaybettikten sonra, hareket kabiliyetlerini geri kazanmak ve yaşamalarını kolaylaştırmak adına protez kullanabilirler. Fakat bazı durumlarda protezler, ampute zamanla hastalara uyum sağlamamakta, hastaların hareket kabiliyetlerini engellemekte ve hastalarda ağrı, acıya sebep olmaktadır. GİPROV; ampute hastaların, protezler ile yaşadıkları sorunların, yapay zeka kullanılarak algılanmasını sağlayan yazılımsal bir bütündür. Görüntü işleme algoritmaları kullanılarak ampute hastanın hareket esnasındaki eklem noktaları ve pozisyonları tespit edilip yapay sinir ağına girdi olarak gönderilmektedir. Daha önceki ampute hastaların verileriyle eğitilmiş olan yapay sinir ağı algoritması, kendisine girdi olarak gelen hasta eklem ve pozisyon bilgileri ışığında hastanın protez ile olan uyumunu ve sorunlarını tespit etmektedir. Tüm sistem mobil ve bilgisayara uyumlu uygulama sayesinde daha ulaşılabilir hale getirilmiştir. Yapay sinir ağına doktorlar tarafından onaylı veri seti ile uygulanan testler ve ampute hastaların projenin işlevselliği hakkındaki görüşleri ışığında GİPROV'un ampute hastalar için oldukça yararlı olduğu sonucuna varılmıştır. Sonuç olarak GİPROV adı ampute hastalar için hayat kolaylaştırıcı ve gelişime açık bir projedir.



# **1. Amaç**

## **1.1. Problem**

Ampute hastalarda bulunan protezler ile ilgili sorunlar ikiye ayrılmaktadır.

### **1.1.1. Zaman İçerisinde Meydana Gelen Problemler**

Fiziksel gelişimi fazla olan ampute hastaların (çocuk vb.) ampute edilmiş bölgelerindeki değişim, protezlerin yerinin değişmesine ve hastaların hareket kabiliyetinin kısıtlanmasına sebep olmaktadır. Ayrıca protezlerde zamanla meydana gelen aşınma, kayma gibi sorunlarda hastaların günlük yaşamına olumsuz etki etmektedir. (Turner Shruti, 2022, #)

### **1.1.2. Başlangıç Temelli Problemler**

Hastaya uygun protez tasarılanırken süreç esnasındaki çeşitli hatalar, yanlış ve hastanın hareket kabiliyetlerini kısıtlayıcı protezlerin üretilmesine sebep olmaktadır. Bu durumda protezler hastalara uyum sağlayamamakta ve hastada ağrı, acı, dolaşım bozukluğu ve yaralanma meydana gelmesine sebep olmaktadır.



Bütün bu sorunlar sonucunda protezindeki problemleri fark edemeyen hastalar günlük hayatı kısıtlayıcı problemlerle karşılaşmakta, sorunları fark eden hastalar doktorlara başvurmakta bu durum ise hastaneler, doktorlar üzerinde bir yoğunluk ve iş yükü oluşturmaktadır. Ayrıca sağlık sistemindeki yoğunluklar sebebiyle hastalar, protezdeki sorunların tespit edilmesi için bir süre beklemekte bu süreç içerisinde protezler işlevini yerine getirememektedir. (Turner Shruti, 2022, #)

## **1.2. Projenin Amacı**

GİPROV'un amacı, ampute hastaların protezleriyle ilgili problemlerini; yapay zeka ve görüntü işleme kullanarak, pratik ve ulaşılabilir bir şekilde tespit etmek böylece doktor ve hastanelere düşen iş yükünü azaltıp ampute hastaların hayatlarını kolaylaştırmaktır.

## **1.3. Hedef Kitle**

GİPROV, ampute hastaların protezleriyle ilgili olan problemlerini tespit etmeyi amaçladığından ampute hastaları hedeflemektedir. Ayrıca GİPROV, protez sorunu tespit sürecini kısaltmak için doktorlar tarafından kullanılabilir. GİPROV'da yer alan uygulama sayesinde tüm sistem telefonda ve bilgisayarda kolayca bulundurulabilir.

## **1.4. Ekonomik Veriler Ve Verimlilik**

GİPROV, tamamen yazılımsal ve yapay zeka temelli olduğundan herhangi bir maliyeti yoktur. Yapılan testler ve veriler ışığında ise sistemin ampute hastalar için oldukça faydalı olduğu sonucuna ulaşılmıştır.

# **2. Yenilik Unsurları Ve Literatür Taraması**

## **2.1. Literatür Taraması**

2009 yılında elektromiyografi (EMG) verilerini denetleyen ve el protezlerinin verimliliğini artırmaya yarayan bir sistem kurulmuştur (Todd A. Kuiken, 2009, #). Fakat elektromiyografi verileri düşük ölçüm doğruluk oranlarına sahip olduğundan çalışmanın sonucunda beklenilen efektifliğe ulaşlamamıştır. Ayrıca sistemin sadece el protezleri için işlevsel olması çalışmanın kapsamını azaltmıştır. Yapay zeka ile protez verimliliğini artırmaya yönelik birçok çalışma olsada mevcut protezin sorunlarını ve ampute hastaya uyumunu tespit edebilen bir proje olmamıştır.

2012 yılında hastalardan elektromiyografi (EMG) verileri alınarak ve bu veriler yapay zeka ile işlenerek kişilere uygun el protezi tasarımları yapılmış ve anlık el protezi kontrolü yapılmıştır (Giulia C Matrone, 2012, #). Fakat elektromiyografi verileri düşük doğruluk oranlarına sahip olduğundan çalışma sonucunda istenilen verimlilik alınamamıştır. Ayrıca elektrmiyografi sensörlerinin kullanılmış olması çalışmanın erişilebilirliğini düşürmüştür, protezin hastaya uyumluluğu ve verimliliği tespit edilememiştir.

2019 yılında giyilebilir sensörler çeşitli insanlara giydirilmiş ve belirli bir yürüyüş periyodu yapmaları istenmiştir (Sasanka Potluri, 2019, #). Ardından sensörlerden gelen veriler yapay zeka ile işlenmiş ve kişilerin sahip olduğu yürüyüş bozuklukları tespit edilmiştir. Fakat verilerin alınabilmesi için sensörlerin kullanılması gerektiğinden çalışmanın erişilebilirliği azalmıştır. Ayrıca sensörler nabız verilerini kullanarak ölçüm yaptığından, sistem protezler için yeterli olamayacaktır.

2022 yılında gerçek zamanlı bir şekilde insan eklem noktalarının pozisyon bilgilerini tespit edebilen bir sistem yapılmıştır (Poongavanam Palani, 2022, #). Fakat bu sistem protezleri algılayamamaktadır. Vücut eklem noktalarını görüntü işleme ile algılayan birçok sistem bulunmaktadır fakat bu sistemler aynı zamanda protezleri algılayamamakta ve elde edilen tüm verileri sürece göre listeleyememektedir.

## **2.2. Literatür Özeti**

Protez sektörünün teknoloji ile birleşimi ilk etapta elektromiyografi (EMG) verileri ile protez kontrolüne ve hareketine dayanmaktadır. Ardından elektromiyografi (EMG) verileri yapay zeka kullanılarak işlenmiş ve ampute hastaya uyumlu protez tasarımları yapılmıştır. Giyilebilir teknoloji ve yapay zeka teknolojilerinin ilerlemesiyle tüm vücuta takılı sensörler kullanılarak kişilerin yürüyüş anormallikleri tespit edilmiştir. Fakat giyilebilir sensörler nabız verisine ihtiyaç duyduğundan bu sistem protezler üzerinde etkili olamamıştır. Görüntü işleme teknolojilerinin ilerlemesi ile insan vücudunun eklem bölgeleri ve pozisyonları kamera aracılığıyla tespit edilebilir hale gelmiş ve bu alan üzerinde çeşitli çalışmalar yapılmıştır. Fakat bu sistemler protez algılama konusunda ve süreç içerisindeki eklem pozisyonlarını listelemekte verimsiz olmuştur.

## **2.3. Yenilik Unsurları Ve Literatürdeki Boşluklar**

Görüntü işleme teknolojisini kullanarak ampute hastanın eklem noktalarını, eklem noktalarının pozisyon ve açı bilgilerini, protezin konum ve açı bilgilerini tespit eden bir çalışma bulunamamıştır.

Görüntü işleme sisteminde gelen verileri yapay sinir ağları ile işleyerek protezin hastaya sorunlarını ve hastaya olan uyumluluğunu, verimliliğini tespit eden bir çalışma bulunamamıştır.

Tüm bu sistemleri tek bir uygulamada birleştiren bir çalışma bulunamamıştır.

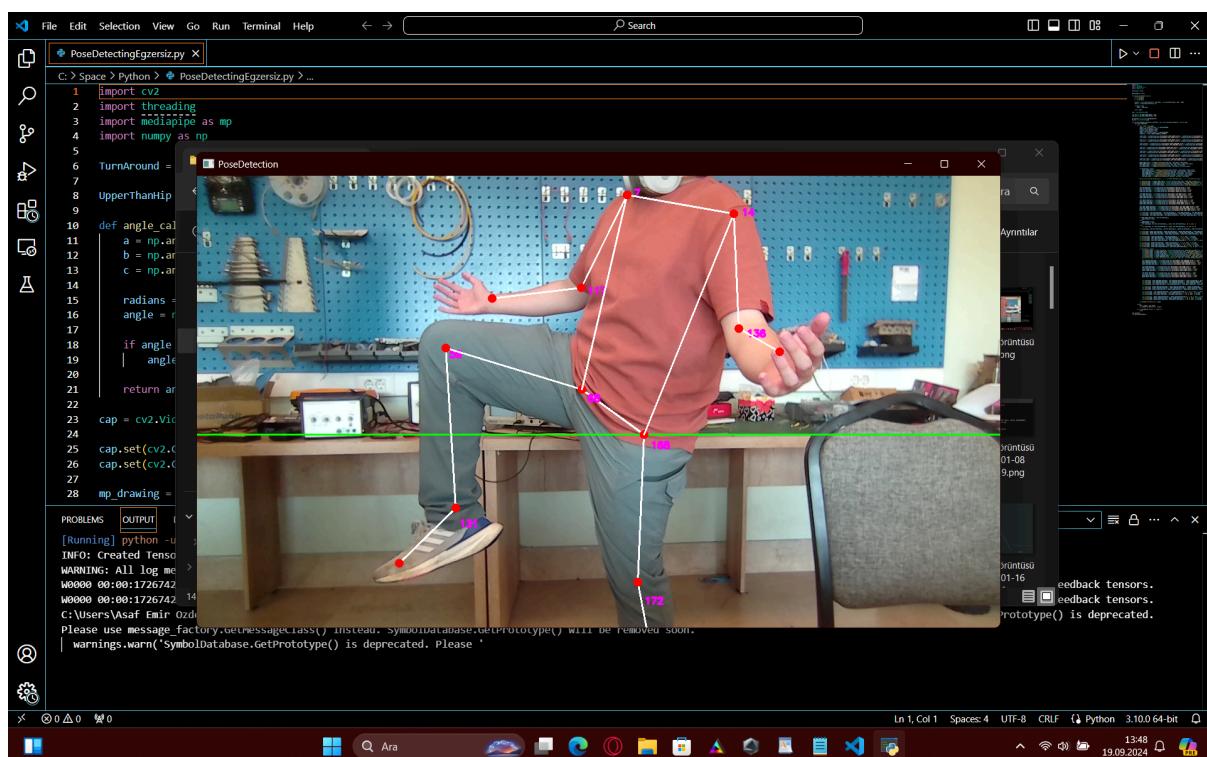
Tespit edilen literatür boşlukları ve GİPROV'un literatüre getirmiş olduğu yenilik unsurlarına bakıldığından GİPROV'un özgün ve yenilikçi bir sistem olduğu sonucuna ulaşılmıştır.



## 3. Yöntem

### 3.1. Görüntü İşleme

Ampute hastanın anlık eklem pozisyon verilerini alabilmek için bir görüntü işleme algoritması oluşturulmuştur. Algoritma hastanın eklemlerinin pozisyon ve rotasyon bilgilerini saniyede 10 kere tespit etmekte ve elde edilen bilgileri zaman akışına göre listelemektedir. Daha sonrasında elde edilen veriler işlenmek üzere yapay sinir ağının algoritmasına gönderilmektedir.



### 3.2. Yapay Sinir Ağısı

#### 3.2.1. Algoritma

Yapay sinir ağısı sisteminin kurulması için matematiksel bir algoritma oluşturulmuştur. Tüm sistem kodları bu algoritma üzerine kurulmuştur.

##### 3.2.1.1. Yapay Sinir Ağısı'nın Temel Değişkenleri

Sistemde kullanılan yapay sinir ağısı 10 katmandan oluşmaktadır. Her katmanda sırasıyla; 16, 32, 32, 64, 64, 128, 64, 32, 32, 16 adet nöron bulunmaktadır. Öğrenme oranı olarak 0.01 kullanılmıştır. Ağırlıklar ise -3.1 ve 3.1 arasındaki rastgele değerlerle başlatılmıştır. Tüm yapay sinir ağısı algoritması C++ dili ile kodlanmıştır.

### 3.2.1.2. Sigmoid Aktivasyon Fonksiyonu

Yapay zeka algoritmasının ilerlemesiyle, nöronların değerleri artmaktadır. Bu sebeple algoritmanın ilerleyen süreçlerinde de verileri saklayabilmek için lineer olmayan bir fonksiyon kullanılmaktadır. Tüm sistemde aktivasyon fonksiyonu olarak sigmoid kullanılmıştır.

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad \sigma'(x) = \frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$$

```
((1 / (1 + exp(-x -hidden[i] / sigmoidConstant))) *  
 (1 - (1 / (1 + exp(-x -hidden[i] / sigmoidConstant))))) *  
 ((double)(1.0 / (double)sigmoidConstant)) * (double)learningRate;
```

### 3.2.1.3. Yapay Sinir Ağı Zincir Kuralları

Her bir ağırlığın (weight) ve her bir nöronun (hidden unit) değerlerinin, sistemin çıkışının etkileme oranı zincir kuralı ile sağlanır.

$$\begin{aligned}\frac{\partial f(g(x))}{\partial x} &= \lim_{h \rightarrow 0} \frac{f(g(x+h)) - f(g(x))}{h} \\&= \lim_{h \rightarrow 0} \frac{f(g(x+h)) - f(g(x))}{h} \cdot \frac{g(x+h) - g(x)}{g(x+h) - g(x)} \\&= \lim_{h \rightarrow 0} \frac{f(g(x+h)) - f(g(x))}{g(x+h) - g(x)} \cdot \frac{g(x+h) - g(x)}{h} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x}\end{aligned}$$

### 3.2.1.4. Hidden Unit'lerin Değişim Değerinin Bulunması

Çıktı nöronlarının hata oranı  $Loss_n = \frac{1}{2} \cdot (Observed_n - Predicted_n)^2$  formülü ile bulunur. (Observed: olması gereken değer, Predicted: sistemin tahmin ettiği değer). Belirli bir nöronun değişim değerinin bulunması için, o nöronun değerinin Loss'a göre parçalı türevine bakılır.

(S: katman sayısı, C(i): i'nci katmandaki hidden unit sayısı,  $h_{i,j}$ : i'nci katmandaki j'nci nöron,  $w_{i,j,k}$ : i'nci katmandaki j'nci nörondan i+1'nci katmandaki k'ncı nörona giden ağırlık,  $\alpha$ : öğrenme oranı)

## Çıktı Nöronları İçin

$$\frac{\partial Loss_i}{\partial h_{S,i}} = \frac{\partial Loss_i}{\partial \sigma(h_{S,i})} \cdot \frac{\partial \sigma(h_{S,i})}{\partial h_{S,i}} = (Observed_i - Predicted_i) \cdot \sigma'(h_{S,i}) \cdot \alpha = \Delta h_{S,i}$$

## Ara Nöronlar İçin

$$\begin{aligned} \frac{\partial Loss_n}{\partial h_{i,j}} &= \frac{\partial Loss_n}{\partial \sigma(h_{S,n})} \cdot \frac{\partial \sigma(h_{S,n})}{\partial h_{S,n}} \cdot \frac{\partial h_{S,n}}{\partial \sigma(h_{S-1,n})} \cdot \frac{\partial \sigma(h_{S-1,n})}{\partial h_{S-1,n}} \cdots \frac{\partial \sigma(h_{i,j})}{\partial h_{i,j}} \\ &= \sigma'(h_{i,j}) \cdot \left( \sum_{k=0}^{C(i+1)} w_{i,j,k} \cdot \Delta h_{i+1,k} \right) = \Delta h_{i,j} \end{aligned}$$

```
... global__ void backPropagationStart(double* hidden, double* dH, double* input, double* dB) {
    int i = threadIdx.x;

    dH[i] = (input[i] - hidden[i]) * ((1 / (1 + exp(-hidden[i] / sigmoidConstant))) *
    [(1 - (1 / (1 + exp(-hidden[i] / sigmoidConstant)))) * ((double)(1.0 / (double)sigmoidConstant)) * (double)learningRate;
    dB[i] = dH[i];
}
```

### 3.2.1.5. Ağırlıkların (Weight) Değişim Değerinin Bulunması

Ağırlıkların hata nöronunu bulmak için aynı Loss Fonksiyonu kullanılır. Bir ağırlığın değişim değerini bulmak için o ağırlığın etki ettiği nöronlara göre ve Loss'a göre parçalı türevine bakmak gereklidir.

(S: katman sayısı, C(i): i'nci katmandaki hidden unit sayısı,  $h_{i,j}$ : i'nci katmandaki j'nci nöron,  $w_{i,j,k}$ : i'nci katmandaki j'nci nörondan i+1'nci katmandaki k'ncı nörona giden ağırlık,  $\alpha$ : öğrenme oranı)

$$\begin{aligned} \frac{\partial Loss_n}{\partial w_{i,j,k}} &= \frac{\partial Loss_n}{\partial \sigma(h_{S,p})} \cdot \frac{\partial \sigma(h_{S,p})}{\partial h_{S,p}} \cdot \frac{\partial h_{S,p}}{\partial \sigma(h_{S-1,p})} \cdot \frac{\partial \sigma(h_{S-1,p})}{\partial h_{S-1,p}} \cdots \frac{\partial \sigma(h_{i,j})}{\partial h_{i,j}} \cdot \frac{\partial h_{i,j}}{\partial w_{i,j,k}} \\ &= \sigma'(h_{i,j}) \cdot \Delta h_{i+1,k} = \Delta w_{i,j,k} \end{aligned}$$

```
... global__ void backPropagateWeights(double hidden[][maxHidden], double dw[][maxWeight], double dH[][maxHidden], int index) {
    int i = threadIdx.x;
    int j = threadIdx.y;

    /*switch (index) {
        case 0 :
            dw[index][i * blockDim.y + j] = hidden[index][i] * dH[index + 1][j];
        default :
            dw[index][i * blockDim.y + j] = ((1 / (1 + exp(-hidden[index][i] / sigmoidConstant))) *
            (1 - (1 / (1 + exp(-hidden[index][i] / sigmoidConstant)))) * ((double)(1.0 / (double)sigmoidConstant)) * dH[index + 1][j];
    }*/
    if(index == 0) {
        dw[index][i * blockDim.y + j] = hidden[index][i] * dH[index + 1][j];
    } else {
        dw[index][i * blockDim.y + j] = ((1 / (1 + exp(-hidden[index][i] / sigmoidConstant))) *
        (1 - (1 / (1 + exp(-hidden[index][i] / sigmoidConstant)))) * ((double)(1.0 / (double)sigmoidConstant)) * dH[index + 1][j];
    }

    dw[index][i * blockDim.y + j] *= learningRate;
}
```

### 3.2.1.6. Çıktı Verme

Yapay sinir ağı “Feed Forward” adındaki algoritmayı kullanarak çıktı verebilmektedir. Bu algoritmaya göre ;

$$h_{i,j} = \sum_{n=0}^{C(i-1)} w_{i-1,n,j} \cdot h_{i-1,n}$$

### 3.2.1.7 Algoritma Sonucu

Tüm bu işlemlerin sonucunda; yapay sinir ağı önce veri setine bağlı kalarak çeşitli ampute durumlarında hastaların hareket esnasında olması gereken eklem pozisyonlarını öğrenir. Ardından algoritma; girdi olarak verilen, hastanın hareket esnasındaki eklem pozisyon bilgilerini inceleyerek protezin sorunlarını ve hastaya olan uyumunu, verimliliğini tespit eder.

El Bileği: %88, Dirsek: %17 Verim oranına sahiptir. Protezin Dirsek bölümünün verimi beklenenden az olduğundan Dirsek bölgesinde sorun tespit edilmiştir.

Algoritmadan alınan örnek çıktı.

### 3.2.2. Cuda İle Ekran Kartını Kullanarak Algoritmayı Çalıştırmak

Yapay sinir ağı algoritması oldukça matematiksel işlem gerektirmektedir. Bu sebeple tüm matematiksel işlemleri doğrudan işlemcide çalıştırmak algoritmanın eğitim süresini ve bilgisayarın iş yükünü artırrır. Bu sebeple algoritmadaki tüm işlemler, çekirdek sayısı bilgisayar işlemcisinden 300 kat fazla olan ekran kartında aynı anda yapılmaktadır. Bu sayede algoritmanın eğitim süresi ve bilgisayara bıraktığı iş yükü oldukça azalmaktadır. Tüm bu sistemleri ekran kartı üzerinde çalıştmak için ise Nvidia'nın üretmiş olduğu Cuda kütüphanesi kullanılmıştır. Not: Algoritmalar ve algoritmik kodlar tamamen özgündür.

## 3.3. Kullanılan Uygulamalar Ve Yazılımlar



**Visual Studio** : Tüm sistemin kodlarını ve algoritmalarını yazmak için kullanılmış editördür.



**C++** : Yapay sinir ağı algoritmalarını kodlamak için kullanılmış yazılım dilidir. Hızlı olması en önemli avantajlarından biridir.



**Python** : Görüntü işleme algoritmasını kodlamak için kullanılmış yazılım dilidir. Pratik olmasına ön plana çıkmaktadır.



**Nvidia Cuda** : İş yükünden ve zamandan tasarruf etmek için; yazılan algoritmaları, ekran kartında çalıştırılabilmek için kullanılmış C++ kütüphanesi.



**Media Pipe** : Python ile görüntü işleme algoritmasını yazmak için kullanılmış kütüphane.

### 3.4 Veri Seti

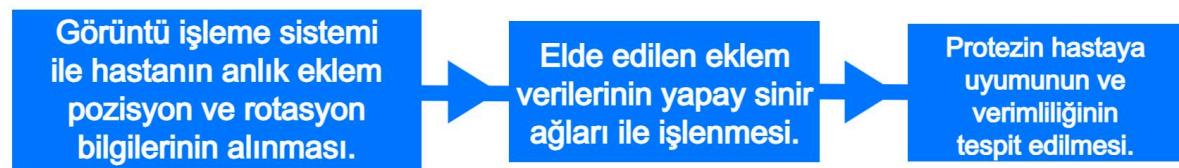
Çeşitli ampütyasyonlara sahip hastaların 16 adet eklem noktasının rotasyon ve pozisyon bilgileri, 1.6 salise aralıklarla kaydedilmiştir ve 1 milyon 700 bin insanın verisi toplanmıştır. Ayrıca tüm hastaların protezlerinin olması gereği gibi sorunsuz çalıştığı doktorlar tarafından onaylanmış, hastaların ampütyasyon çeşitleri veri setine kaydedilmiştir. Bu sayede yapay sinir ağları ampute çeşitlerine göre eklemelerin olması gereken rotasyon ve pozisyon bilgilerini öğrenmiş ve kendisine gelen verileri doğru olanlarla kıyaslayarak protez sorunlarını bulabilir hale gelmiştir. Tüm veri seti “Medeley Data” açık kaynaklı ve doktor onaylı veri paylaşım platformundan alınmıştır. (Gerekli bağlantılar kaynakçada bulunmaktadır).

```
patient_id,date,walking bout_id,datetime,foot,stride length [m],stride time [s],swing time [s],stance time [s],gait speed [m/s],ic angle [deg],fc angle [deg],max. foot lift [m]
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:09.308593750,right,1.707673241655287,1.1328125,0.44921875,0.68359375,1.5074632753922537,-12.13990241934276,41.68872957939712,0.1150077204045118
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:09.865234375,left,1.7700877785232485,1.162109375,0.41015625,0.751953125,1.5231679707628625,-11.181184213846025,40.91120179135305,0.1184962195345455
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:10.480468750,right,1.673825635333166,1.2109375,0.46875,0.7421875,1.3822560085333682,-2.3748344645533166,40.648397326236,0.1113725887471288
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:11.623046875,right,1.6152145673847478,1.15234375,0.46875,0.68359375,1.401677726272866,-0.6642444056361541,40.10826117019212,0.027830201711319
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:12.179687500,left,1.681187646387881,1.171875,0.419921875,0.751953125,1.4346134582509915,-10.739564095774869,36.61207536131317,0.0955702173555913
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:12.775396025,right,1.70642749363129,1.15234375,0.498846875,0.654296875,1.481192351989683,-4.75807926472689,43.09338293883437,0.098818734762718
s-a17fb78a,2021-02-28,011eff8,2021-02-28 12:40:13.322265625,left,1.894199697282711,1.162109375,0.4296875,-12.49894590426453,41.322703201874646,0.108391720262274
```

⋮

### 3.5. Algoritma Akış Şeması

Kullanılan algoritmanın tüm süreç boyunca oluşturduğu akış şeması aşağıda verilmiştir.



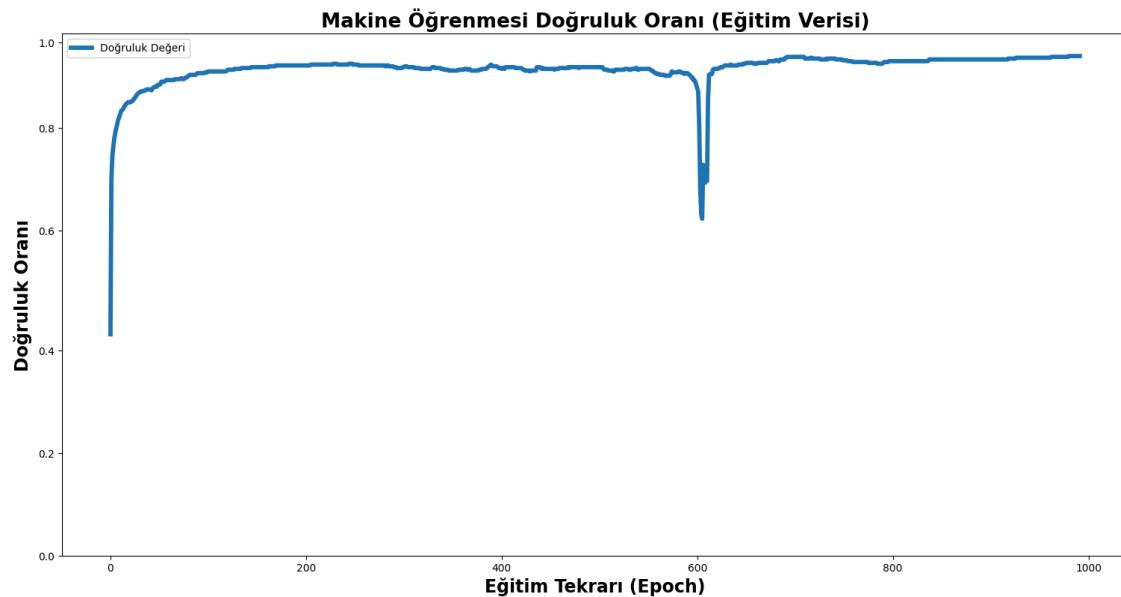
## 4. İş, Süreç Ve Planlama

| İşler                              | Aylar |       |         |        |         |       |  |
|------------------------------------|-------|-------|---------|--------|---------|-------|--|
| Ve Görevler                        | Nisan | Mayıs | Haziran | Temmuz | Ağustos | Eylül |  |
| Literatür taraması                 | X     | X     | X       | X      | X       | X     |  |
| Yazılımların geliştirilmesi        |       |       | X       | X      | X       |       |  |
| Saha çalışmaları                   | X     |       |         |        | X       | X     |  |
| Verilerin toplanması ve analizleri |       |       | X       | X      | X       |       |  |
| Proje raporu yazımı                |       |       |         | X      | X       | X     |  |

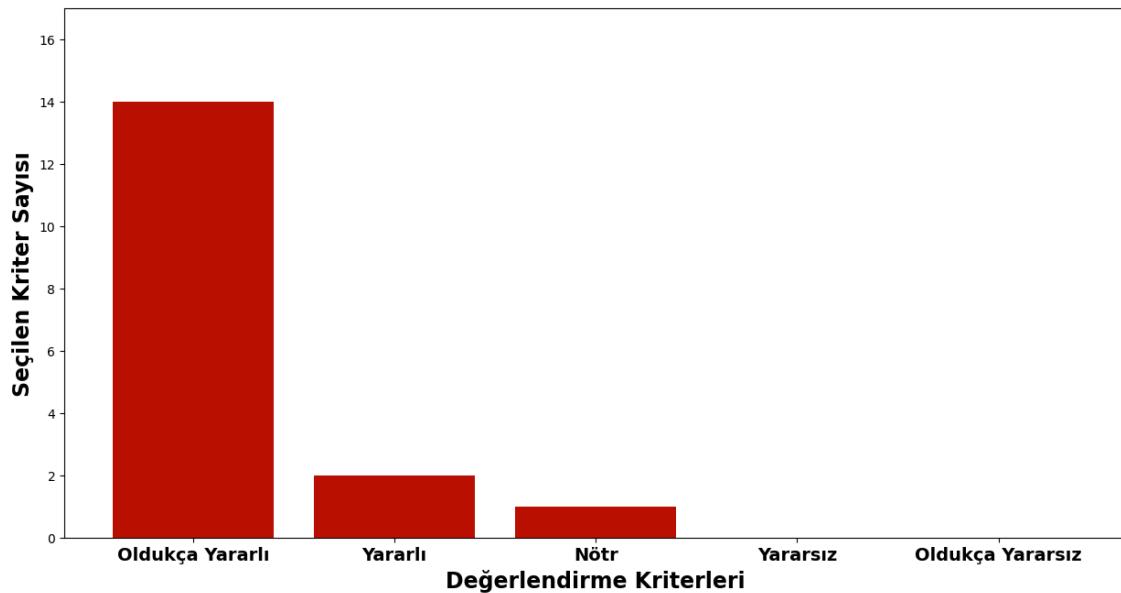
Tablo 1: Aylara göre yapılan işler tablosu.

## 5. Sonuç

### 5.1. Bulgular



Tablo 2 : Eğitim süresine (tekrarına) göre makine öğrenmesi (yapay sinir ağı) çıktı doğruluk oranı.



Tablo 3 : 17 kişilik ampute hasta grubuna yapılan anket sonuçları.

### 5.2. Analizler

Yapay sinir ağı 1000 tekrar boyunca eğitilmiş ve her tekrar sonucunda sistemin doğruluk oranı test edilip elde edilen veriler Tablo 2'ye yazılmıştır. Elde edilen verilere göre yapay sinir ağıının eğitim süresi (tekrarı [epoch]) arttıkça elde edilen çıktı doğruluk oranları da artmaktadır. Eğitim tekrarının en fazla olduğu noktada çıktı doğruluk oranı %88.7'dir.

17 adet ampute hastaya projemizi anlatan bir metin ve projemizi değerlendirmeleri için bir anket gönderilmiştir. Anket sonucunda elde edilen değerler Tablo 3'e aktarılmıştır. Tablo 3'teki verilere göre anket katılımcılarının %82'si GİPROV'u oldukça yararlı, %11'i yararlı, %7'si ise nötr olarak değerlendirmiştir.

### **5.3. Sonuç Ve Tartışma**

Elde edilen bulgu ve bulgu analizlerine göre GİPROV ampute bireyler tarafından oldukça yararlı görülmektedir. Ayrıca sistemden elde edilen çıktıların doğruluk oranları literatürdeki çalışmalara göre yüksektir. Literatürde protezin sorunlarının, uyumluluğunun ve verimliliğinin tespit edilmesi ile ilgili boşluklar ve GİPROV'un literatüre kattığı bazı yenilikler göz önünde bulundurulduğunda, GİPROV ampute hastaların protezleriyle ilgili sorunlarını, protezin verimliliğini ve uyumluluğunu tespit edebilmekte ve ampute hastaların hayatlarını kolaylaştırmaktadır.

## **6. Geliştirilebilirlik**

- Görüntü işleme sisteminin çalıştırıldığı bilgisayarın gücü arttırlarak saniyede daha fazla verinin elde edilebilmesi sağlanabilir.
- Yapay sinir ağının çalıştığı ekran kartı gücü arttırlarak daha fazla katmanlı ve daha fazla nöronlu algoritmalar oluşturulabilir.
- Kullanılan veri setinin kapsamı arttırlarak daha yüksek doğruluk oranlarına sahip algoritmalar oluşturulabilir.
- Daha fazla ampute birey ile görüşülerek projenin eksik ve geliştirilebilir yönleri tespit edilebilir.

## **7. Öneriler**

- Görüntü işleme yerine Convolutional yapay sinir ağları kullanılarak bilgisayarın iş yükü azaltılabilir.
- Tüm sistem Transformer (Tensor) olarak yeniden kodlanıp çok daha yüksek doğruluk oranları elde edilebilir.
- Geliştirilen mobil uygulama daha fazla platform için çıkartılarak sistemin ulaşılabilirliği artırılabilir.
- GİPROV, E-nabız'a entegre edilerek sistemin daha erişilebilir olması sağlanabilir.
- Sistemin, kuantum veya analog bilgisayarlarda derlenmesi sağlanarak doğruluk oranı ve kapsamı çok büyük oranda arttırılabilir.

## Kaynakça

Aziz, Mazi H. “Automated Body Postures Assessment From Still Images Using Mediapipe.” *DergiPark*, 2023, <https://dergipark.org.tr/tr/download/article-file/3071793>.

“CUDA C++ Programming Guide.” 2024, [https://docs.nvidia.com/cuda/pdf/CUDA\\_C\\_Programming\\_Guide.pdf](https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf).

Kavi, Krishna. “Multithreading Implementations.” *Research Gate*, 1998, [https://www.researchgate.net/publication/2734966\\_Multithreading\\_Implementations](https://www.researchgate.net/publication/2734966_Multithreading_Implementations).

Kuiken, Todd A. “Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms.” *PubMed*, 2009, <https://pubmed.ncbi.nlm.nih.gov/19211469/>.

Matrone, Giulia C. “Real-time myoelectric control of a multi-fingered hand prosthesis using principal components analysis.” *Journal of NeuroEngineering and Rehabilitation*, 2012, <https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-9-40>.

Palani, Poongavanam. “Real-time Joint Angle Estimation using Mediapipe Framework and Inertial Sensors.” *IEEE XPLOR*, 2022, <https://ieeexplore.ieee.org/abstract/document/9973559>.

“Partial Derivative (Definition, Formulas and Examples) | Partial Differentiation.” *BYJU'S*, <https://byjus.com/math/partial-derivative/>. Accessed 20 September 2024.

Potluri, Sasanka. “Deep Learning based Gait Abnormality Detection using Wearable Sensor System.” *IEEE XPLOR*, 2019, <https://ieeexplore.ieee.org/document/8856454>.

Sakryukin, Andrey. “Under The Hood of Neural Networks. Part 1: Fully Connected.” *Medium*, 2018, <https://towardsdatascience.com/under-the-hood-of-neural-networks-part-1-fully-connected-5223b7f78528>.

Shruti, Turner. “Issues faced by people with amputation(s) during lower limb prosthetic rehabilitation: A thematic analysis.” *Journals*, 2022, [https://journals.lww.com/poijournal/fulltext/2022/02000/issues\\_faced\\_by\\_people\\_with\\_amputation\\_s\\_during\\_12.aspx](https://journals.lww.com/poijournal/fulltext/2022/02000/issues_faced_by_people_with_amputation_s_during_12.aspx).

Vashisht, Manisha. “Role of Mathematics in Image Processing.” 2019, <https://ieeexplore.ieee.org/document/8862438>.

Kullanılan Veri Seti: <https://data.mendeley.com/datasets/k5y9jkx87y/1>