**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**



**ELE495-BİTİRME PROJESİ RAPORU**

|  |  |
| --- | --- |
| **GÖZ KIRPARAK HABERLEŞME** | |
| **Yılmaz Çınar** | **161201037** |
| **Mesut Demircioğlu** | **161201011** |
| ***Raporun Teslim Tarihi*** | ***20.05.2020*** |

**YAZ 2020**

**İçindekiler**

**1.Teori..........................................................................................................................3**

**2.Literatür Araştırmaları...........................................................................................3**

**3.Yöntem......................................................................................................................3**

**4.WBS, İş bölümü ve zaman planlaması...................................................................4**

**5.Referanslar ...............................................................................................................5**

**6.Youtube Linki...........................................................................................................6**

# 1.TEORİ

ALS hastaları için düşünülmüş bu projede, hastalığın belli evrelerinde en etkili kullanabildikleri kas olan göz kapakları ile hastaların iletişim kurabilmesi amaçlanmaktadır. Oluşturulacak projenin olabildiğince stabil ve kullanıcının genel ihtiyaçlarına cevap verebilmesi hedeflenmektedir. Genel olarak ana düşünce, kullanıcının yüzünün görüntüsünü bir kamera ile kaydedip, gerçek zamanlı olarak işlemci üzerinde işlemektir ve kelimesi bittiğinde bu kelimenin bir LCD ekran yardımı ile çıktısının alınmasıdır.

# 2.LİTERATÜR ARAŞTIRMALARI

ALS hastaları iletişim sırasında zorluk çeker. Hastaların yaklaşık olarak %75'i iletişimde yardıma ihtiyaç duyar. Yapılan araştırmalara göre %28'i tamamen konuşamazken %47'si konuşma sırasında zorluk çeker. Sadece %25'i anlaşılabilir şekilde konuşabilir. ALS'ye sahip bireylerde dizartri adı da verilen çevresel ve merkezi sinir sistemindeki hasarların yol açtığı konuşmayı sağlayan kasların bozulması durumu görülmektedir. Hastaların iletişimde kullandığı yöntemler ikiye ayrılmıştır. Bunlar teknolojisiz ve teknolojiyle iletişimdir. Teknolojisiz iletişimde bir kere göz kırpma evet, iki kere göz kırpma hayır demektir. Teknolojinin kullanıldığı birçok iletişim yöntemi vardır. Bu projede kullanılacak olan yöntem göz kırpma ile haberleşmedir. Göz kırpma ile haberleşme yapılması için ilk olarak gözün tespit edilmesi gerekir. OpenCV kütüphaneleri yardımıyla gerçek zamanlı şekilde göz tespit edilebilir. Göz kırpmanın anlaşılması için yapılan algoritmalardan biri göz en boy oranı(EAR)'dır. Gözün 6 farklı noktaları için koordinatlar hesaplanır. Koordinatlar ile yapılan işlemler sonucu belli bir eşik değerinin altındaysa göz kapalı durumda demektir. EAR sonucunun eşiğin altına kaç kere düştüğü hesaplanırsa göz kırpma sayısı bilinir. Bizim kullanacağımız algoritma da bu algoritmayla benzerlik gösterir.

# 3.YÖNTEM

Teknik açıdan bu projede kullanılacak işlemci ve çevre birimleri şunlardır:

* Raspberry Pi Model 3B+
* PiCamera v2
* 16x4 LCD Ekran A1604
* WF Tripod WT3110A

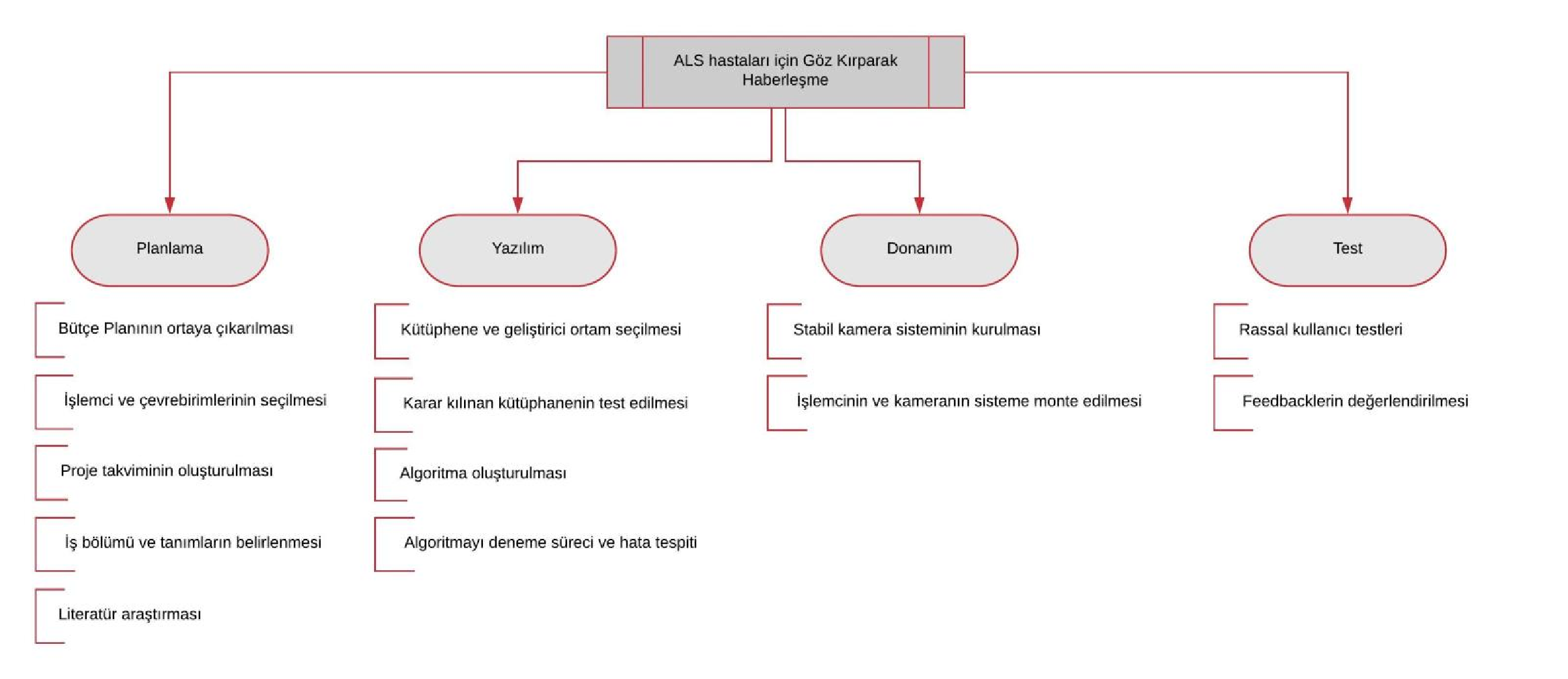
Raspberry Pi modelleri ile uyumlu olduğu için seçtiğimiz kameramız ile kullanıcın yüzü sürekli olarak kayıt altına alınacaktır. İşlemci birimimize(Raspberry Pi) kablo ile bu veriler aktarılacaktır. Burada gerçek zamanlı olarak alınan bu veriler Python3 dili aracılığı ile ve OpenCV kütüphanesi kullanılarak işlenmektedir. Bu görüntü işleme yönteminden kısaca bahsedecek olursak; öncelikle yüzün tespit edilmesinin gerekliliğinden dolayı bir cascade filter her frame’e uygulanmaktadır.

Ardından da yine bir cascade filtre yardımı ile gözün konumu belirlenip kontrol amaçlı olarak işlenmiş video görüntüsü ekrana yazdırılmaktadır. Buradan sonra ise elimizdeki verinin işlenme sürecine geçiş yapılıp işlenen veri LCD ekrana yazdırılmaktadır.

Şu ana kadar belli aşamalarda karşılaştığımız bir FPS sorunu ile uğraştık. Çözme aşamasında olduğumuz bu sorun bizim bazı göz kırpmalarını kaçırmamıza yol açıyordu. Kodu yeterince sadeleştirerek ve işlemcinin ısınma sorununu çözerek üstesinden gelebileceğimizi düşünüyoruz.

Ayrıca zaman ve maaliyet el verirse projeye eklemek istenilen eklentiler de mevcuttur. Bunlardan biri, çıktıyı sadece LCD ekran ile değil aynı zamanda önceden kaydedilmiş sesleri bir hoparlör yardımı ile ortama vermektir.

*WBS*



# 4. WBS, İŞ BÖLÜMÜ VE ZAMAN PLANLAMASI

Her projede olduğu gibi bizim projemizde de iş bölümü yapmanın ve belli bir zaman çizelgesine uymaya çalışmanın gerekliliğinden dolayı, aşağıda gördüğünüz zaman çizelgesi ve WBS şemasını hazırladık.

*Zaman Çizelgesi*

|  |  |
| --- | --- |
| 27 Nisan - 5 Mayıs | Literatür araştırmaları |
| 5 Mayıs - 12 Mayıs | Bütçe planlanması ve malzeme siparişlerinin tamamlanması |
| 12 Mayıs - 20 Mayıs | Yazılımın hazrılanıp ilk testlerin yapılması ve ilk gelişme raporunun sunulması |
| 20 Mayıs - 27 Mayıs | Feedbacklerin değerlendirilip yazılım iyileştirilmesi |
| 27 Mayıs - 3 Haziran | İkinci gelişme raporunun hazırlanıp sunulması |
| 3 Haziran - 10 Haziran | Son testlerin yapılıp demoya projenin hazır hale getirilmesi |
| 10 Haziran - 17 Haziran | Final sunumu ve posterinin hazırlanıp, son demosunun sunulması |

Pandemi dolayısıyla genelde internet konferansıyla şu ana kadar sürdürdüğümüz bu projede iki taraf da aynı anda literatür araştırması yapıp sonuçları eş zamanlı olarak tartışmıştır. Algoritma geliştirme süreci de eş zamanlı olarak video konferanslarıyla devam etmiştir. Belli bir zamandan sonra araştırma kısmında Mesut Demircioğlu ve test kısmında da Yılmaz Çınar çalışmalarını yoğunlaştırmıştır.

# 4. REFERANSLAR

* *Saunders C, Walsh T, Smith M. Hospice Care in the Motor Neuron Diseases.*

*London: Edward Arnold, 1981.*

* *Brownlee A, Palovcak M, The Role of Augmentative Communication Devices in the Medical Management of ALS. Journal of NeuroRehabilitation, Volume 22, Number 2, 2007, 445-447.*
* *Tereza Soukupova and Jan Cech , Real-Time Eye Blink Detection using Facial Landmarks, 21st Computer Vision Winter Workshop, Rimske Toplice,*

*Slovenia, February 3–5, 2016*

# 5. YOUTUBE LİNKİ

https://youtu.be/-OHk3LcsFOg