

**Yerel Voksel Ağları İle Beyin Verisi Üzerinden Bilişsel Süreçlerin Modellenmesi Ve Otomatik Olarak Tanınması**

**Program Kodu: 1001**

**Proje No: 112E315**

Proje Yürütücüsü:  
**Prof. Dr. Fatoş T. YARMAN VURAL**

Araştırmacı:  
Doç. Dr. İlke ÖZTEKİN

Bursiyerler:  
Emre AKSAN  
Burak VELİOĞLU  
Arman AFRASİYABİ  
Sarper ALKAN

## ÖNSÖZ

Bu projede, makine öğrenme ve nörobilim yöntemleri harmanlanarak yeni bir beyin modeli geliştirilmiştir. **Yerel Voksel Ağları (YBA)** adını verdiğimiz bu model farklı bilişsel süreçler sırasında kaydedilen Fonksiyonel Manyetik Rezonans Görüntüleme (fMRG) sinyallerini kullanarak her bir bilişsel süreç için bir beyin ağı üretir. Elde edilen beyin ağı değişik amaçlar için kullanılabilir: Nörobilimciler, proje kapsamında geliştirilen görselleme aracı ile beyin ağını inceleyebilir ve çeşitli çıkarımlarda bulunabilirler. Geliştirilen beyin ağının kenar ağırlıklarını kullanarak bir sınıflandırıcıyı eğitebilir ve bilişsel süreçlerin otomatik olarak tanınması mümkün olabilir. Böylece, ileriki yıllarda, beyin sinyallerini doğrudan kullanabilen insan-makina ara yüzleri tasarlamak mümkün olabilecektir.

Yerel Voksel Ağları (YVA), insan beyninin yerellik ve dağıtıklık özelliklerini bir arada kullanarak bir dizi doğrusal modelin bir araya getirilmesi ile elde edilmiştir. Geliştirdiğimiz bu model, projenin nihai amacı olan bilişsel süreçleri otomatik olarak tanıma ve etiketleme konusunda literatürde kullanılan yöntemlere göre daha başarılı performanslar ortaya koymaktadır.

TÜBİTAK tarafından desteklenen proje kapsamında geliştirilen bu model ve sonuçları, ulusal ve uluslararası yayınlar ve hazırlanan yazılımlar ile benzer konularda araştırma yapan bilim insanlarının hizmetine sunulmuştur. Proje kapsamında 3 adet yüksek lisans tezi ve 12 adet bilimsel yayın gerçekleştirilmiştir. Bu desteği bizlere sağladığı ve ülkemizin bu yeni alandaki birikimlerini desteklediği için TÜBİTAK-EEEAG ' ye teşekkür ederiz. fMRG verileri Bilkent Üniversitesi UMRAM merkezinde yapılan uzun, yorucu ve kapsamlı deneyler ile elde edilmiştir. Destekleri için UMRAM' a da teşekkürlerimizi sunuyoruz.

## İçindekiler

1. Giriş.....	9
2. Literatür Özeti.....	9
3. Proje Kapsamında Geliştirilen Yöntemler .....	11
3.1. Deneilerin Tasarlanması Programlanması ve Verilerin Elde Edilmesi.....	11
3.2. Görüntü İşleme Teknikleri ile Verilerin Analiz Edilmesi ve İyileştirilerek ÇVÖA Analizine Hazır Hale Getirilmesi.....	13
3.3. Deney Verileri Üzerinde ÇVÖA Analizi ile Mevcut Makine Öğrenme Algoritmalarının Uygulanması ve Sonuçların Değerlendirilmesi .....	20
3.4. İnsan Beyninde Bağlanırlık.....	21
3.5. Yerel Voksel Ağları Yöntemi (YVA): .....	22
3.7. Algoritmaların Paralleleştirilmesi ve Optimizasyonu.....	35
3.8. CEREBRA: Proje Kapsamında geliştirilen görselleme aracı ve kullanıcı arayüzü ...	40
3.9. Hazırlanan Yazılımların Web Tabanlı Bir Paket Haline Getirilmesi .....	44
4. Sonuç.....	45

## Şekil Listesi

Şekil 1 – Tanıma Testi Deney Şablonu.....	12
Şekil 2 – İki farklı nesne sınıfına karşılık gelen resim örnekleri .....	13
Şekil 3 – Tekrardan yön verme işleminin öncesi ve sonrası.....	16
Şekil 4 – Yeniden hizalandırma .....	17
Şekil 5 - Anatomik görüntü üzerine "Coregistration" işlemi uygulandıktan sonra elde edilen görüntü.....	18
Şekil 6 - Voksel $v(t_i, s_j)$ ve yerel komşuluğunu gösteren ağ diyagramı, en yakın 4 komşu ile oluşturulmuştur.....	23

## Tablo Listesi

Tablo 1 – 2. Deney türüne ait veriler üzerinde farklı sınıflandırıcı performansları.....	21
Tablo 2 - 2. Deney türünden elde edilen veriler için ham veri kullanılarak eğitilen ve test edilen 4 ayrı sınıflandırıcı performansları. ....	32
Tablo 3 – 2. Deney türünden elde edilen veriler için DVM kullanılarak elde edilen (%) sınıflandırma performansları.....	33
Tablo 4 – 2. Deney türünden elde edilen veriler için DVM kullanılarak elde edilen (%) sınıflandırma performansları.....	34

## ÖZET

Yapay zeka konusunda çalışan arařtırmacılar, insan zekasından esinlenerek, insan zekasına benzeyen yapay sistemleri geliřtirmeye çalışmaktadır. Temel amaç insan gibi düşünen, insan gibi öğrenen ve problem çözebilen makinalar yapabilmektir. Bu sistemlerle ilgili yazılım ve donanımları geliřtirebilmek üzere birçok matematiksel yöntem ve algoritmalar geliřtirilmiřtir. Bu yöntemler sayesinde nesnelere algılama, tanıma, sınıflama ve öğrenme gibi önemli biliřsel süreçlerin matematiksel modelleri oluşturulmuřtur.

Bu projede, yapay zeka algoritmaları için geliřtirilen yöntemleri kullanarak, biliřsel süreçleri (örneğin bellek, öğrenme, duygulanım) modelledik. Diđer bir deyiřle, bugüne kadar insan zekasını taklit ederek geliřtirdiđimiz yapay zeka yöntemlerini tersinir bir mühendislikle geri çevirerek dođal zekayı modellemek üzere kullandık. Modellerimizi biliřsel süreçler sırasında kaydededilen fonksiyonel Manyetik Resonans Görüntüleri (fMRG) kullanarak oluşturduk. Bunun için, bir dizi biliřsel deney tasarladık ve bu deneyleri denekler üzerinde uygularken fMRG sinyallerini kaydettik. Böylece, elde ettiđimiz etiketli öğrenme verilerini geliřtirdiđimiz makine öğrenme algoritmalarını eğitmek için kullandık.

Projenin en önemli çıktılarından birisi de oluşturulan modellerin ve elde edilen fMRG verilerinin, web tabanlı bir ortamda tüm arařtırmacıların hizmetine açılmasıdır. Böylece oluşturduğumuz metodoloji ve programları nörobilimciler veri analizinde kullanabilecekler ve kendi fMRG verilerini de modelleyebileceklerdir.

Proje kapsamında geliřtirdiđimiz matematiksel beyin modeline **Yerel Voksel Ağları (YVA)** adını verdik. İnsan-bilgisayar etkileřimi teknolojileri için bilimsel bir altyapı oluřturma potansiyeline sahip olan bu model, iki farklı veri seti üzerinde gerçekteřtirilen deneylerde, literatürde yoğun olarak kullanılan diđer Çoklu Voksel Örüntü Analizi (ÇVÖA) yöntemlerine göre daha başarılı sonuçlar ortaya koyduđu anlařılmıřtır.

**Yerel Voksel Ağları** yöntemi, fMRG sinyallerinin en küçük birimi olan voksellerden elde edilen zaman serileri arasındaki iliřkiyi dođrusal denklemlerle modellemektedir. İnsan beyinde, birbirine yakın nöronların benzer aktiviteler gösterdiđi bilinmektedir. Bu yerel benzerlik, birbirine yakın vokseller arasındaki iliřkinin dođrusal olduđunu göstermektedir. Bazen de birbirinden uzak vokseller de beyindeki direk yollar aracılıđı ile benzer aktivasyonlara sahip olabilmektedir. Bu tür vokseller yerel olarak deđil de fonksiyonel olarak birbirine komřu kabul edilebilir. Geliřtirmiş olduđumuz bu yeni komřuluk sistemine fonksiyonel komřuluk adını verdik. Ve vokseller arasındaki dođrusal iliřkiyi, "Fonksiyonel Komřuluk" adını verdiđimiz bu ölçütü kullanarak modelledik. **YVA** modelinin, fMRG sinyallerinden gelen ham özniteliklere göre, biliřsel süreçleri etiketleme konusunda daha başarılı olduđunu deneysel olarak gösterdik.

Proje kapsamında öncelikle fMRG verisi toplamak üzere bir dizi deney tasarladık ve bu deneyleri Bilkent Üniversitesi UMRAM merkezinde gerçekleştirdik. Bunun için insan beyninin bilgileri nasıl depolandığını ve nesnelere nasıl tanıdığını anlamak üzere 3 değişik deney yaptık. Bu deneylerde değişik nesnelere tanındığı etiketli fMRG verileri elde ettik. Daha sonra verileri işleyerek gürültülerden arındırmaya çalıştık ve modelimiz için uygun hale getirdik. Kullandığımız görüntü iyileştirme teknikleri ile, **YVA** yönteminde elde edilen etiketleme performansını arttırdık.

**YVA** yöntemi sonucunda elde edilen beyin ağını kullanarak birçok öznelik oluşturulabilir. Biz bu çalışmada **YVA**' dan elde ettiğimiz kenar ağırlıklarını öznelik olarak kullanarak Yapay Sinir Ağları, Destek Vektör Makinaları ve k-En Yakın komşuluk gibi çeşitli sınıflandırıcıları eğittik. Bunların içinde en başarılı olanları seçerek performansları ölçtük.

Yöntemin başarılı sonuçlar vermesi neticesinde, doğrusal ilişki özneliklerinin çıkarılma adımının hızlandırılması için GPU programlama tekniklerinden faydalandık. Son olarak da ortaya çıkarılan özneliklerin beyin modeli üzerinde bilim insanlarına sunulması için, proje kapsamında bir kullanıcı ara yüzü geliştirdik. Böylece, oluşturduğumuz beyin ağlarını bilim insanlarının kullanımına sunmayı hedefledik.

## ABSTRACT

Researchers are highly inspired from the human brain to develop intelligent algorithms. The main goal of these algorithms is to build a machine which is capable of learning and solving problems as human do. Numerous mathematical models and algorithms have been developed to enhance the software and hardware of these intelligent systems. Recently, well-established mathematical models have been created for object detection, recognition and classification tasks.

In this project, we modeled the cognitive states, e.g. memory, learning, emotion, using methods developed for intelligence systems. In other words, we employed the models in artificial intelligence, which are developed by imitating the human intelligence, to model the human brain itself. The proposed models are based on the functional magnetic resonance imaging data (fMRI) recorded during different cognitive processes. In order to gather these data, we have also designed cognitive experiments and applied them on participants. Therefore, we have used the labeled dataset to train machine learning algorithms suggested in the project.

The most important outcome of the project is that all the models and programs are shared with other researchers through the web. Thereby, developed methodologies and programs can be used by other neuroscientists to analyze and model their own data.

Our proposed model, called “**Local Mesh Model**”, has a potential to be a basis for human-computer interaction systems. After the experiments done on two different datasets, it is observed that Local Mesh Model gives better results than heavy-duty MVPA methods used in the literature.

“**Local Mesh Model**” method assumes a linear interaction between the local voxels, which are the smallest unit of fMRI signals. It is known that, locally close voxels have similar time series for the same types of stimuli. The local similarity indicates a linear relation between nearby voxels. It is also known that, some voxels in the brain have similar activation patterns with distant voxels due to the direct links between some groups of neurons. These voxels can be accepted as functional neighbours. We define a new neighborhood system, called functional neighborhood and define functional meshes to model the linear interactions among the voxels in the same functional neighborhood. Experimental results obtained for “**Local Mesh Model**” established in functional neighborhood is more successful than raw fMRI intensity values to label cognitive states.

In the project, we first design a set of cognitive experiments. Then we record the fMRI data in Bilkent University UMRAM research center. Three different experiments are applied to understand how the information is stored and object is recognized in the brain. Labeled fMRI data are obtained while participants discriminating different objects. After the raw data is

recorded, image processing techniques are applied to denoise the data and make them proper for our models. Results show us that these techniques help us to improve our labelling performance.

Using the brain network obtained by estimating the “Local Mesh Model”, we can define many different features. In this project, we used arc weight of these network as a feature to train Neural Network, Support Vector Machines and k-NN classifiers. We used the most successful classifiers to measure performances of our models.

We also improve the speed of model creation using GPU programming techniques. As a final contribution, we developed a user interface to serve our models and features to other researchers. We aim to share created brain network with other researchers.



## 1. Giriş

Proje kapsamında gerçekleştirilen 11 adet iş paketinden ilk 5 iş paketi fMRG verilerinin elde edilmesi için gerekli deneylerin tasarlanmasını, bu tasarımlara göre deneylerin gerçekleştirilerek verilerin toplanmasını ve elde edilen verilerin bazı görüntü iyileştirme teknikleri ile gürültülerden arındırılmasını içermektedir. Böylece, fMRG veri seti Çoklu Voksel Örüntü Analizi (ÇVÖA) için uygun hale getirilmiştir.

6,7,8 ve 9 numaralı iş paketlerinde ise, geliştirmiş olduğumuz Yerel Voksel Ağları (YVA) modelinin istatistiksel olarak geçerliliği analiz edilmiş ve bilişsel süreçleri tanımadaki başarısı araştırılmıştır. **YVA** modeli ile eğitilen sınıflandırıcılardan elde edilen performanslar literatürdeki benzer yöntemlerle karşılaştırılmış ve elde edilen performansların mevcut yöntemlere göre daha iyi olduğu gözlemlenmiştir.

10 ve 11 numaralı iş paketleri kapsamında ise, oluşturulan yöntem ve algoritmalar ile elde edilen sonuçların benzer alanda çalışmalar yapan diğer bilim insanları ile paylaşılması amaçlanmıştır. Bu amaç doğrultusunda sonuçların görselleşebileceği bir kullanıcı ara yüzü geliştirilmiştir. Kendi veri setleri üzerinde geliştirilmiş algoritmaları koşturmak isteyen bilim insanları için, geliştirilen **YVA** yöntemi bir MATLAB paketi olarak araştırmacıların hizmetine sunulmuştur. Proje kapsamında geliştirilen tüm algoritma ve programlar <http://neuro.ceng.metu.edu.tr> sitesinde bilim insanlarının hizmetine sunulmuştur.

## 2. Literatür Özeti

İnsan zihnini araştırırken karşılaştığımız en zor ve en önemli sorunlardan biri matematiksel modelleme problemidir. Bilgiler beynimizde nasıl temsil edilir? Bu temsiller bilginin tipi ve doğasına göre nasıl değişir? Bu soru bilim tarihi boyunca birçok disiplinin karşı karşıya geldiği merak uyandırıcı ve bir o kadar da zorlayıcı bir problem olmayı sürdürmüştür.

Araştırmacılar zihinsel durum ve süreçleri davranış ve bilişsel süreçleri analiz etmek için fMRG verilerini değerlendirmektedirler. Son yıllarda makine öğrenme ve nörobilim alanlarında geliştirilen yöntemler bize bilişsel süreçler sırasında beyinde süregelen işlemleri yansıtan nöral verilere erişimimizi sağlamıştır. Bu gelişmeler, insan zihnini incelemeyi hedefleyen birbirini bütünleyici birçok yeni yöntem ve bakış açılarının oluşumunu beraberinde getirmiştir (Norman, Polyn, Detre ve Haxby, 2006; Polyn, Natu, Cohen ve Norman, 2005; Wang, Hutchinson ve Mitchell, 2003; Haynes ve Rees, 2006; Mitchell, Hutchinson, Niculescu, Pereira, Wang, Just ve Newman, 2003) [1, 2, 3, 4, 5]. Bu yöntemler içinde en önemlilerinden biri Çoklu Voksel Örüntü Analizi (ÇVÖA) adı verilen yöntemdir. ÇVÖA yönteminde, fMRG ile ölçülen beyin aktiviteleri makine öğrenme yöntemleri kullanılarak modellenir. Bu amaçla, elde edilen etiketli fMRG verileri kullanılarak beyindeki dağılmış beyin aktivite dizinleri bir istatistiksel öğrenme (makine öğrenmesi) algoritması ile modellenir. Bu aktivite dizinlerinden her dizine karşı gelen

bilgi, ya da temsil tipini tanımlamaya ve birbirinden ayırt etmeye çalışılır. Bu açılım ile ÇVÖA, yukarıda belirtilen temsil problemi sorusunu endirekt bir şekilde cevaplayabilecek yegane yöntemdir. ÇVÖA ile FMRG verisi bir makine öğrenme algoritmasına girdi olarak verilerek, manipule edilen bilgi tipleri veya bilişsel süreçlerin beyin verisi baz alınarak tanımlanıp tanımlanamadığı ve de en önemlisi birbirlerinden başarı ile ayırt edilip edilemeyeceği test edilebilir.

ÇVÖA kullanan mevcut yaklaşımlarda, FMRG verilerinden elde edilen voksel ölçümleri voksel sayısı uzunluğunda bir vektör altında birleştirilerek Yapay Sinir Ağları, Destek Vektör Makinaları, Bayes sınıflandırıcılar gibi yaygın olarak kullanılan öğrenme sistemlerini eğitirler. Literatürde bu alanda yapılmış bazı çalışmalar mevcuttur. Bu çalışmalar aşağıda özetlenmektedir: De Martino, Valente, Staeren, Ashburner, Goebel ve Formisano, 2008 [6] yılında yaptıkları çalışmada voksel değerlerinin oluşturduğu vektör uzayını doğrusal olarak bölütleyen Destek Vektör Makinası kullanmışlardır. Benzer olarak, Jin, Strasburger, Laken, Kozel, Johnson, George ve Lu, 2009 [7] yılında yaptıkları çalışmada Fisher yöntemi kullanmışlardır. Ayrıca, Sun, Morris, Lee, Taylor, Mills ve Babyn (2010) [8] zaman serileri yöntemi ile modelledikleri vokselleri kümelemişlerdir. Michael, Gramfort, Varoquaux, Eger, Keribin ve Thirion, (2011) [9] yine Destek Vektör Makinaları ile beyindeki bazı zihinsel durumları sınıflandırmışlardır. Wang, Shi, Yeung, Tsang ve Heng, (2007) [10] yeni bir Destek Vektör Makinası geliştirmiş ve bu makine ile bazı beyin aktivitelerini kümelemiştir. Chu, Ni, Tan, Saunders ve Ashburner (2011) [11], Kernel Regresyonu ile beyin durumlarını modellemişlerdir. Cabral, Silverira ve Figueriredo, (2011) [12] Bayesian ve K- en yakın komşuluk algoritmasını birleştirerek beyin verisini kodlamıştır. Plumpton, Kuncheva, Oosterhof ve Johnston, (2011) [13] birden fazla öznelik uzayını birleştirerek lineer bir sınıflandırıcı kullanmıştır ve gerçek zamanlı hızlı bir algoritma geliştirmiştir. Kuncheva ve Rodriguez (2011) [14] bazı sınıflandırıcıları deneysel olarak karşılaştırmıştır.

Proje kapsamında ortaya çıkarılan **YVA** yönteminde ise, fMRG verisinden gelen ham voksel sinyallerini direk kullanmak yerine, voksel sinyalleri arasındaki doğrusal ilişkilerden elde edilen ağırlıklar makine öğrenmesi sınıflandırıcılarına girdi olarak verilmektedir. Çalışma kapsamında, detayları “Kaynaklar” kısmında verilen, ulusal ve uluslararası pek çok yayın yapılmıştır. **YVA** yönteminin çıkış makalesi olarak kabul edilebilecek olan Ozay (2011) [15] vokseller arasındaki doğrusal ilişkileri, bölgesel olarak yakın voksellerin komşuluk sınırları içinde hesaplamış ve elde edilen doğrusal ilişkilerin öznelik olarak kullanılmasının bilişsel süreç etiketleme konusunda ham veriye oranla daha başarılı sonuçlar ürettiğini ortaya çıkarmıştır. Fırat (2013) ve Fırat (2014) [16, 17] makalelerinde, komşuluk seçiminde fonksiyonel benzerlik kullanımının, bölgesel komşuluk seçimine göre daha başarılı sonuçlar ürettiği gösterilmiştir. Önal (2014-1, 2014-2) [18, 19] makalelerinde ise doğrusal ilişkilerin çıkarılmasında kullanılan komşu sayısının bilişsel süreç etiketlemedeki öneminin üzerinde

durulmuş ve bilgi teoretik yöntemler ile elde edilen optimal komşu sayısının etiketleme performansını arttırdığı gözlemlenmiştir. Velioğlu (2014) [20] çalışmasında, enformasyon kuramlı (Information Theoretic) yöntemlerle oluşturulan komşuluk beyinde daha önceden belirlenmiş olan anatomik bölgeler ile karşılaştırılmış ve oluşturulan modelin biyolojik olarak ne kadar tutarlı olduğu belirlenmeye çalışılmıştır. Önal (2015) [21] birden çok çekim anının verisini doğrusal ilişki çıkarımı yaparken ortak kullanarak, elde edilen doğrusal ilişki özniteliklerinin kalitesini arttırmayı amaçlamaktadır.

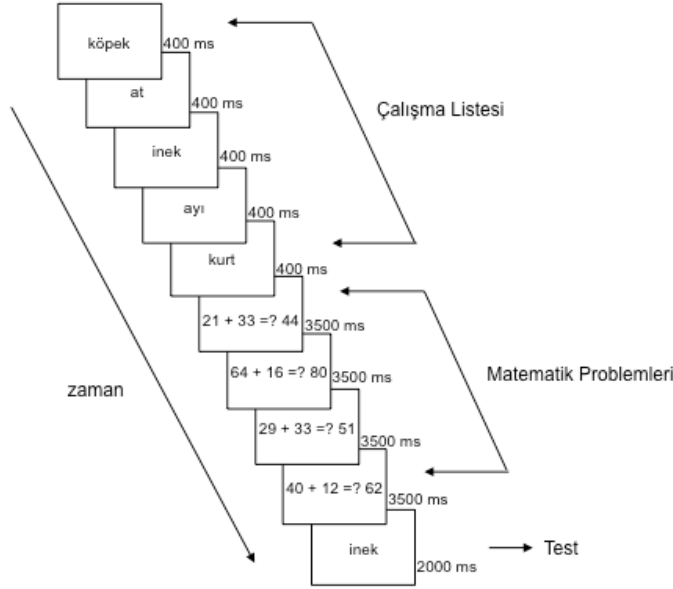
### 3. Proje Kapsamında Geliştirilen Yöntemler

Proje kapsamında gerçekleştirilen bütün çalışmalar proje önerisinde belirtilen iş paketlerinde yapılan araştırmalar halinde detaylandırılacaktır.

#### 3.1. Deneylerin Tasarlanması Programlanması ve Verilerin Elde Edilmesi

Bu projede kullanılan tüm veri setleri proje ekibimiz tarafından öğrenci denekler kullanılarak UMRAM merkezinde toplanmıştır. Proje kapsamında aşağıda detaylarıyla açıklanan üç deney tasarlanmış, fMRG çekimlerine uygun bir şekilde programlanmış ve fMRG verileri toplanmıştır. Üç deney için de Matlab üzerinde çalışan **Psychophysics** programlama tabanı kullanılmıştır. Bu taban kullanılarak deneylerin fMRG çekimleri ile uyumlu bir şekilde hazırlanmasını takiben veri toplanmaya başlanmıştır. Deneylerle ilgili tasarım ve uygulamalar aşağıda özetlenmektedir:

**Deney 1:** İlk deneyde bir nesne tanıma görevi kullanılarak belleğe bilgi kaydı ve bellekten bilgiye erişim sırasında işlenen bilgi tipine ilişkin beyin aktivitelerinin modellenmesi hedeflenmiştir. Bu çerçevede Şekil 1'de örneği gösterilen tanıma testi deneyi gerçekleştirilmiştir. Bu deneyde katılımcı aynı kategoriye ait (örneğin hayvanlar), ekranda her biri 400 milisaniye gösterilen beş kelime çalışır. Bu çalışma listesini takiben 14 saniye boyunca matematik problemleri çözülür. Matematik problemleri sırasında ekranda gösterilen problemin sonucunun doğru olup olmadığı konusunda yanıt verilir. Matematik problemlerini takiben ise bir tanıma test kelimesi ekranda belirir ve 2 saniye süresinde bu test kelimesinin mevcut çalışma listesine ait olup olmadığına dair evet ya da hayır şeklinde yanıt verilir. Şekil 1'de bir örneği gösterilen bu deney düzeneği listedeki kelimeler ve kelimelerin ait olduğu kategoriler değiştirilerek 300 kere tekrar eder.



Şekil 1 – Tanıma Testi Deney Şablonu.

Bu deney düzeneği, belleğe bilgi kaydedilirken ve kaydedilmiş bilgiler bellekten hatırlanırken beyin aktivitelerini ayrı ayrı inceleme ve analiz etmemizi sağlamıştır. Bilgi kaydı ile bilgi hatırlama süreçleri arasında deneye matematik problemi çözdürmedeki amacımız kısa süreli hafızanın çalıştırılması için bir unutturma sürecinin gerçekleşmesini sağlamaktır. Bu unutturma sürecinde elde edilen veriler modelleme sürecinde kullanılmalıdır.

Sunulan listelerdeki kelimelerin aynı kategoriye ait olması ve bu kategorilerin deney süresince değişmesi ise ÇVÖA yöntemleri kullanılarak belleğe bilgi kaydı ve hatırlama sırasındaki nöral aktivitelerin bilgi tipine göre (kelimelerin ait olduğu kategoriler) sınıflandırılmasına izin verebilmiştir. Şöyle ki, kullandığımız makina öğrenme algoritmaları belli bir kategoriye ait beyin aktivite düzeyleri üzerinde eğitilebilmiş ve doğruluk performansları test edilebilmiştir. Toplamda 10 kategori (hayvanlar, renkler, mutfak eşyaları, kimyasal maddeler, giyim eşyaları, meyveler, sebzeler, kokular, vücut organları, mobilyalar) kullanılmış olup, ÇVÖA yöntemlerinin başarı düzeylerinin şans performansına (%10) oranla ne kadar başarılı oldukları irdelenebilmiştir.

**Deney 2:** Bu deneyde nesne tanıma görevi kullanılarak iki farklı nesne tipine ilişkin beyin aktivitelerinin modellenmesi hedeflenmiştir. Bu bağlamda, deney sırasında **kuşlar ve çiçekler** olmak üzere iki tip nesne resmi dörder saniye boyunca ekranda gösterilmiş ve her bir nesne için beyin aktivitelerinin ayrı ayrı incelenebilmesi ve katılımcının dikkat seviyesinin deney

boyunca yükses tutulabilmesi için resimler arasında rastgele sekiz-on-onikişer saniyelik bir dinlenme süresi tanınmıştır. Hem nesne tanıma hem de hafızadan bilgi getirme deneylerini içermesi için deney 1-önceki (1-back) halinde tasarlanmıştır. Yani, deneklerden art arda gösterilen iki resimdeki nesnelere aynı sınıfa aitse bir butona basmaları beklenmiştir. Deney boyunca bu şekilde otuzar kuş ve otuzar çiçek resmine ait beyin aktiviteleri kaydedilmiştir. Yine ÇVÖA yöntemleri kullanılarak, makina öğrenme algoritmaları iki nesne tipine ait beyin aktivitelerini ayırtedebilmeye eğitilmiş ve doğruluk performansları şans düzeyi (%50) ile karşılaştırılabilmiştir. Aşağıda deneyde gösterilen resimlerden birer örnek sunulmuştur.



Şekil 2 – İki farklı nesne sınıfına karşılık gelen resim örnekleri

**Deney 3:** Son deneyde duygu tipine ilişkin beyin aktivitelerinin modellenmesi hedeflenmiştir. Bu bağlamda, International Affective Picture System veri tabanı kullanılarak dünyaca kabul görmüş ve kişilerde pozitif ya da negatif duygular uyandıran resimler gösterilerek bu resimlere ilişkin beyin aktiviteleri kaydedilmiştir. Her bir resim 2. deneyde olduğu gibi dörder saniye boyunca ekranda gösterilmiş ve her resme ilişkin beyin aktivitelerinin ayrı ayrı analiz edilebilmesi için resimler arasında onikişer saniyelik bir dinlenme süresi verilmiştir. Toplamda 60 adet pozitif ve 60 adet negatif duygu uyandıran resim kullanılarak, ÇVÖA yöntemlerinin iki duygu tipine ait nöral aktivite düzeylerini ayırtedebilme performansı test edilebilmiştir. Bu deney tipi için yapılan öngörüler ve devamında yapılan istatistiksel testler neticesinde, elde edilen fMRG verisinin pozitif ve negatif duyguları sınıflandıracak seviyede ayrımcı bilgi içermediği görülmüştür. Bu sebeple, geliştirilen modelleri test etmek için 1. ve 2. deney tipine ait fMRG verileri kullanılmıştır.

### **3.2. Görüntü İşleme Teknikleri ile Verilerin Analiz Edilmesi ve İyileştirilerek ÇVÖA Analizine Hazır Hale Getirilmesi**

Deney çekimlerinden elde edilen fMRG verileri kullanıma hazır değildir. Elde edilen bu ham veriler zaman içinde yapılan taramalar ile her biri yaklaşık 1 mm kalınlığında olan 2 boyutlu dilimler üretilerek elde edilir. Bu dilimler hem zaman kaymalarından dolayı, hem de içerdikleri gürültü açısından kullanılabilir halde değildir. Ayrıca, deneğin ufak tefek hareketleri dahi ek

gürültü kaynakları oluşturur. Yukarıda bahsedilen nedenlerden dolayı, ham veride sinyalin gürültüye oranı çok düşüktür ve bu oranın görüntü işleme ve iyileştirme teknikleri ile yükseltilmesi gerekmektedir.

fMRG makinasından elde edilen ham verilerden bilgi çıkarabilmek için öncelikle gürültü kaynaklarının modellenmesi gerekmektedir. Daha sonra verinin bu modellere göre gürültülerden arındırılması için bir dizi ön işleme adımı uygulanmaktadır. Ön işlemede kullanılan yöntemler sonraki adımlarda elde edilen model parametrelerimizi ve performanslarımızı büyük ölçüde değiştirmektedir. Bilindiği gibi, projemizin temel amacı beyindeki bilişsel süreçleri makine öğrenme yöntemleri ile modellemek ve bu modelleri kullanarak etiketsiz beyin verilerindeki bilişsel süreçleri tanımdır. Tanıma performansını arttırabilmek için fMRG verilerinin ön işleme adımımda çok sayıda deney yapılmıştır. Bu deneylerde çeşitli görüntü işleme yöntemleri denenerek tanıma performansımızı arttırıcı görüntü işleme tekniklerinin ve bu tekniklerle ilgili parametrelerin belirlenmesine çalışılmıştır.

Veri elde edilmesi sırasında kullanılan deney düzeneğiyle elde edilen ham fMRG verileri, SPM (<http://www.fil.ion.ucl.ac.uk/spm/ext/>) isimli bir fMRG veri analizi programı kullanılarak ön işlemeye tabi tutulmuştur. SPM yazılım aracı fMRG görüntülerini işlemek için geliştirilmiş bir görüntü işleme ve iyileştirme yazılımıdır. Bu yazılımda, filtreleme, normalizasyon, eşikleme vb. standart görüntü işleme yöntemlerine ek olarak, 2 ya da 3 boyutlu fMRG görüntülerini işlemek ve görsellemek için özel yazılımlar da bulunmaktadır.

Bu proje kapsamında kullandığımız görüntüler üç boyutlu beyin hacimlerinden oluşmaktadır. Deneğe verilen bir uyarı için yaklaşık her 2 saniyede kaydedilen iki boyutlu dilimler bir araya getirilerek üç boyutlu bir beyin hacmi üretilir. Deney düzeneğine göre değişmekle beraber her bir uyarı için yaklaşık 6-7 beyin hacmi üretilmektedir. Beyin hacimlerinin en küçük birimi vokselidir. Bir voksel yaklaşık 27 mm<sup>3</sup> lük hacim içerir. Bir beyin hacminde kişiden kişiye ve kullanılan görüntü işleme tekniklerine göre değişmek üzere 20 000- 200 000 arasında voksel bulunur. Her bir vokselde ise deney boyunca verilen uyarılara göre değişen zaman serileri yer almaktadır. fMRG verilerinin gürültülerden arındırılması ve normalize edilmesi geliştirilen yöntemin başarısı için büyük bir önem taşımaktadır.

Proje kapsamında kullandığımız fMRG verisini iyileştirdiğimiz görüntü işleme yedi ana başlık altında gerçekleştirilmiştir. Aşağıdaki alt bölümlerde görüntü işleme adımları açıklanmaktadır.

### **3.2.1. DICOM (Tıpta Dijital Görüntüleme ve İletişim) Verilerinin İçe Aktarımı**

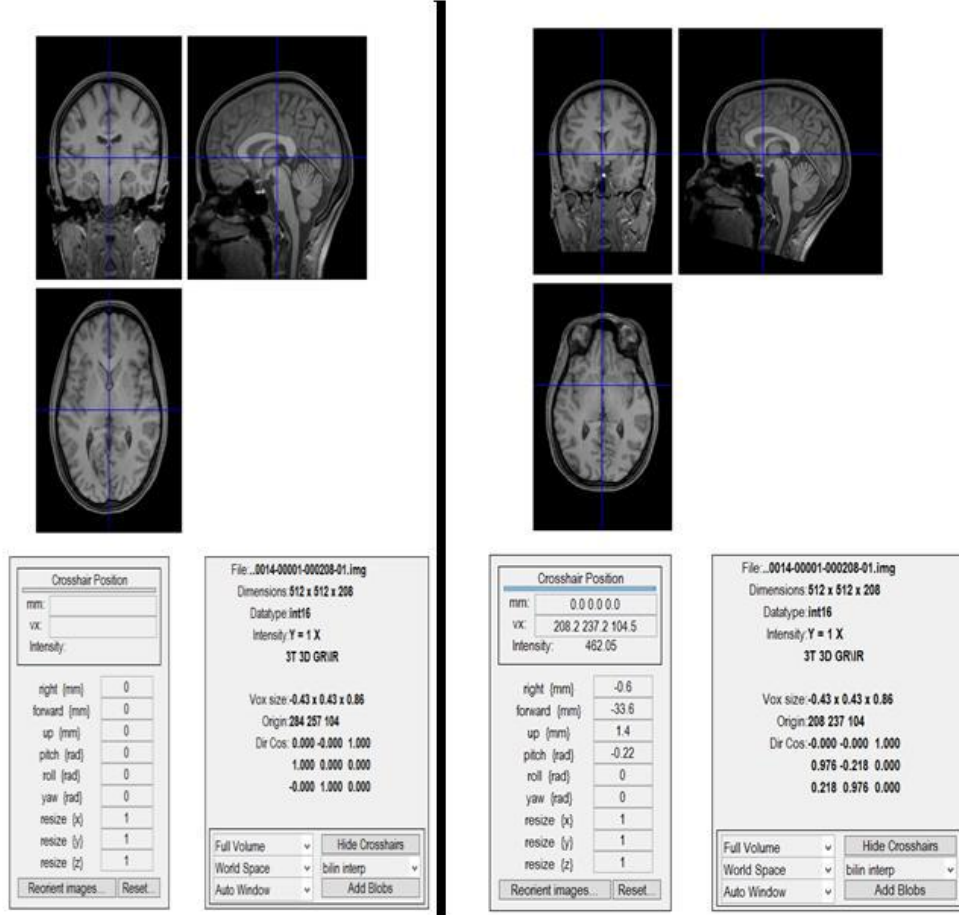
Beyin taramalarını gerçekleştirdiğimiz manyetik rezonans görüntüleme cihazı, verileri DICOM formatında oluşturur. Bu format verileri işlemek için uygun bir format değildir. Görüntülerin iyileştirilmesi için kullandığımız SPM yazılımları elde edilen verilerin tek bir

anatomik görüntü halinde bütünleştirilmesini gerektirmektedir. Bu amaçla DICOM formatındaki veriler SPM yazılımının gerektirdiği formata dönüştürülerek yazılımın içine aktarılır. Bu içe aktarım rutini DICOM dosyalarını SPM uyumlu görüntüler haline getirir. Bu aşamada, öncelikle yapısal veriler birleştirip tek bir anatomik görüntü elde edilmiştir. Daha sonraki aşamalarda ise her seans için ayrı ayrı bulunan veriler kullanılıp fonksiyonel görüntüler elde edilmiştir.

### 3.2.2. Görüntüleri Görselleme

DICOM formatında içe aktarılan veri, üç boyutlu uzayda her biri bir koordinata sahip olan voksellerden oluşmaktadır. Bu voksellerin her biri yaklaşık  $3 \times 3 \times 3 = 27 \text{ mm}^3$  lük bir hacim işgal ederler. Her vokselde deney süresince ölçülen değer bir skaler değer değil, bir zaman serisinden oluşmaktadır. Bu zaman serileri 2 saniyelik aralıklarla ölçülen BOLD (Kandaki oksijen seviyesine bağımlı) sinyaline karşılık gelir.

Bu aşamada kullanılan yazılım bir görüntü hacminin dik kesitlerini etkileşimli bir şekilde görsellemeye yarar. Ekranda solda bulunan panel yardımıyla çeşitli çevirme, döndürme ve zumlama işlemleri uygulanarak görüntülere tekrardan yön verilmiştir. Şekil 3'de tekrardan yön verme işleminin öncesi ve sonrası gösterilmiştir. Burada amaç beynin Anterior Commisure olarak adlandırılan bölgesini orijine yerleştirmektir. Şekilde soldaki görüntüde orijin rastgele bir noktada durmaktadır ama sağdaki görüntüde tam Anterior Commisure'ın üstündedir. Sağdaki şekilde ayrıca, sol alt panelde uygulanan işlemlere ait parametre değerleri gösterilmiştir. "right", "forward" and "up" parametreleri orijini x,y ve z yönünde kaydırmaya yarar. "pitch", "roll" ve "yaw" parametreleri de görüntüleri döndürmek için kullanılır. Sağ alt panel ise görüntü hakkında diğer çeşitli bilgileri gösterir. Bu bilgilerden bazıları şöyledir: görüntü boyutu, veri tipi, görüntü yoğunluk değeri, komşu voksellerin merkezleri arasındaki mesafe, merkezdeki vokselin koordinatları, vb.

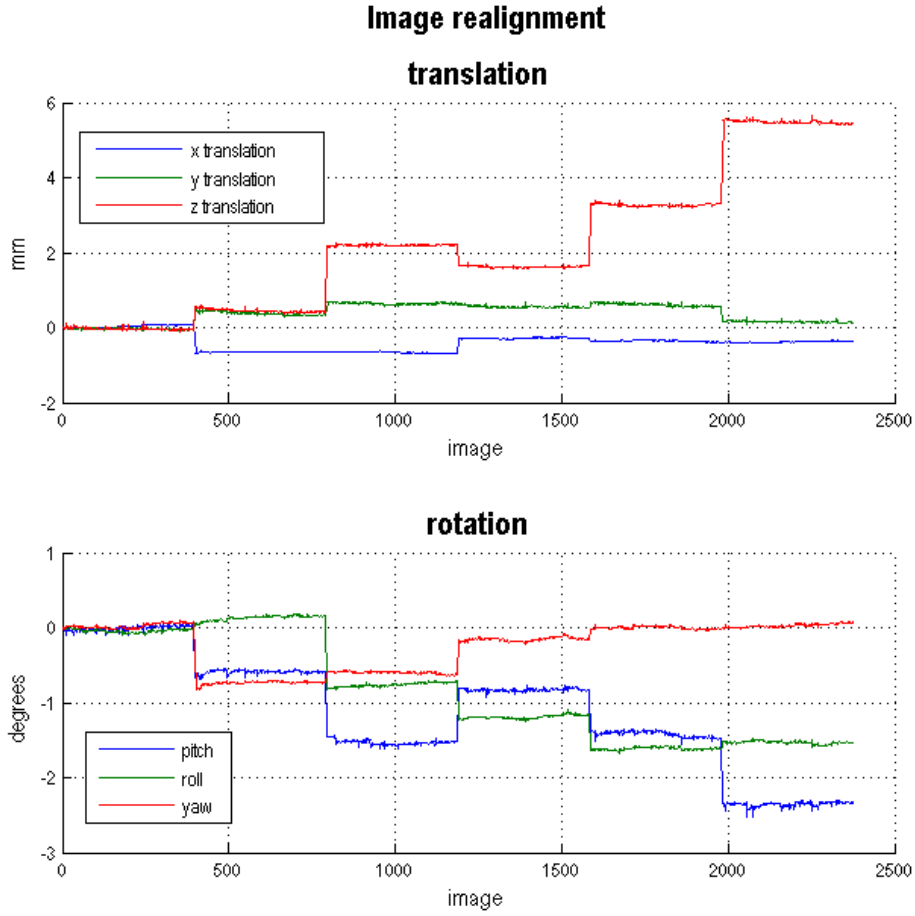


Şekil 3 – Tekrardan yön verme işleminin öncesi ve sonrası

### 3.2.3. Yeniden Hizalandırma

Bu aşamada katılımcının başını oynatmış olma olasılığına karşı elde edilen fonksiyonel görüntüler ilk görüntü referans alınarak hiza edilmiştir. Yeniden hizalandırılmış bu görüntüler daha sonra birleştirilip 3 boyutlu bir görüntü hacmi elde edilmiştir. Şekil 4'te yeniden hizalandırma işlemi esnasında yapılan dönüşümlerin detayı gösterilmektedir. Ötelenmeyi (translation) gösteren grafik katılımcının başını x (sağa sola), y (ileri geri) ve z (aşağı yukarı) yönlerinde ne kadar oynattığını göstermektedir. Grafikteki kırılma noktaları seanslar arası geçişi göstermektedir. Altta grafikteyse (rotation grafiği) denğin 3 farklı şekilde başını ne kadar döndürdüğü görülmektedir. Yukarıda bahsedilen 2 boyutlu fonksiyonel görüntülere bu iki grafikteki deęişim miktarları kadar yeniden hizalandırma uygulanmıştır.



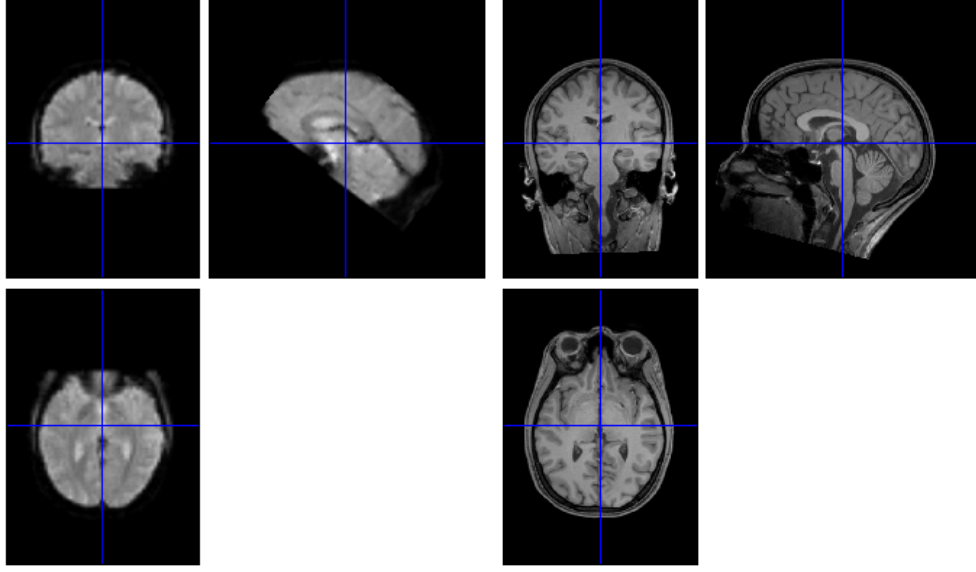


*Şekil 4 – Yeniden hizalandırma*

### 3.2.4. Ortak Kayıtlandırma (Co-Registration)

Bu aşamada DICOM içe aktarma safhasında elde edilen anatomik görüntü ilk seansın yeniden hizalandırılmış ilk fonksiyonel görüntüsü referans alınarak yeniden hizalandırma işlemine tabi tutulmuştur. Aşağıdaki şekilde solda fonksiyonel görüntü ve sağda da anatomik görüntü görülmektedir. Bu ekranda imleç oynatıldığı zaman iki ekranda da birlikte hareket etmektedir. Böylece, anatomik görüntünün fonksiyonel görüntüye hizalanıp hizalanmadığı anlaşılmaktadır.

### Normalised Mutual Information Coregistration



Şekil 5 - Anatomik görüntü üzerine "Coregistration" işlemi uygulandıktan sonra elde edilen görüntü

#### 3.2.5. Bölümlendirme

Sağlıklı bir beyin dokusu genelde 3 ayrı doku tipinde sınıflandırılmaktadır. Bu doku tipleri;

1. Gri madde,
2. Beyaz madde ve
3. Cerebro-spinal sıvı

olarak isimlendirilmektedir. Bölümlendirme işlemi yeniden hizalandırılmış 3 boyutlu anatomik görüntüyü yukarıda belirtilen doku tiplerine ayırmak amacı ile yapılır. Beyinde, bu çalışma kapsamında bilgi taşıyan bölge gri maddenin yer aldığı bölgedir. Amacımız gri maddedeki voksellere ulaşmaktır. Çünkü, en çok aktivasyon bu bölgedeki voksellerde oluşmaktadır. Ayrıca, bu aşama sonucunda bir sonraki aşama olan normalizasyonda kullanılacak parametreler elde edilir.

#### 3.2.6. Normalize Etme

İnsan beyni, yaşa, cinsiyete ve genetik özelliklere göre şekil ve büyüklük bakımından değişiklikler gösterir. Kişiler arası değişimler, bilişsel süreçleri modelleme aşamasında

sorunlara yol açmaktadır. Bu sebepten dolayı elde edilen tüm beyin hacimlerinin standart bir şekle ve büyüklüğe getirilmesi gerekmektedir.

Bu aşamada, birbirinden farklı boyut ve şekilde olan fonksiyonel ve anatomik görüntüler standart bir şablon kullanılarak normalize edilirler. Bu standart beyin şablonunu elde etmek için sinirbilim literatüründe standart sayılan MNI koordinat sistemi kullanılmaktadır. Bu sistem ile elde edilen tüm beyin hacimleri aynı sayıda voksel ile temsil edilir. Montreal Neurological Institute tarafından önerilen ve tüm beyin verilerini aynı standarda geçirmeye yarayan bu koordinat sistemi sarmalama(warping) yöntemi kullanılarak beyin verilerini 3 boyutlu olarak esnetir ve böylece büyüklüklerini sabitler.

İnsan beyninin boyutlarını ve şeklini normalize eden bu sistemin en önemli sorunu, ham fMRG verisindeki voksel sayısını değiştirmesi ve veriyi hafifçe de olsa deforme etmesidir. MNI koordinat sisteminde seçilen voksel boyutuna göre voksel sayıları da değişmektedir. Bu çalışmada voksel boyutları 2 mm, 3mm ve 4mm (x,y ve z) olacak şekilde seçilmiştir. Böylece normalize edilmiş görüntülerde bir vokselin hacmi sırasıyla 8 mm<sup>3</sup>, 27 mm<sup>3</sup> ve 64 mm<sup>3</sup> olmaktadır. Voksel sayısı seçilen voksel boyutu ile ters orantılı olarak değişmektedir. Voksel sayısı arttıkça görüntü orijinale daha yakın olmakla birlikte, makine öğrenme algoritmalarında "boyut laneti" dediğimiz sorundan dolayı geliştirilen modellerin genelleme performansları azalmaktadır.

### 3.2.7. Görüntülerin Pürüzsüzleştirilmesi (Smoothing)

Elde edilen verilerdeki en önemli gürültü kaynağı, tarama esnasında fMRG makinasının çalışma prensiplerinden kaynaklanan artifaktlardır. Bu artifaktlardan fMRG verisini temizlemek için üç boyutlu Gaussian filtreleri kullandık. Gaussian filtrenin yarı maksimumundaki tam genişliğini ("FWHM") ayarlayarak görüntüleri farklı filtreler kullanarak pürüzsüzleştirdik.

Bu aşamada, görüntü değişik FWHM parametresine sahip olan üç boyutlu Gauss filtreleri ile evriştirilir(convolution). Seçilen FWHM parametresine göre daha sonraki aşamalarda uygulanan bölütleme ve tanıma algoritmalarının sonuçları tamamen değişmektedir. Bu parametreler 3 farklı yönde girilir (x,y ve z yönünde) ve birim olarak milimetredir. Biz bu çalışmada, FWHM parametrelerini aşağıdaki formüle göre her yönde eşit olarak belirledik ve deneylerimizi 5 farklı parametre seti kullanarak fMRG verilerini pürüzsüzleştirdik.

$$FWHM = \sigma\sqrt{8\ln 2} .$$

Burada  $\sigma$  standart sapmayı temsil etmektedir.

Kullandığımız Gauss filtrelerinin FWHM parametrelerini literatürdeki çeşitli uygulamalara bakarak belirledik ve sırasıyla şöyle aldık: (mm cinsinden ve x,y,z yönlerinde): [4 4 4], [6 6 6], [8 8 8], [10 10 10], [12 12 12].

### **3.3. Deney Verileri Üzerinde ÇVÖA Analizi ile Mevcut Makine Öğrenme Algoritmalarının Uygulanması ve Sonuçların Değerlendirilmesi**

fMRG verileri kullanılarak beyindeki bilişsel süreçlerin modellenmesi ve tanınması için literatürde Çoklu Voksel Örüntü Analizi (ÇVÖA) adı verilen yöntemler kullanılmaktadır. Çalışma süresince elde edilen deney verilerinin etiketlenmesi konusunda geliştirilen modelin, literatürdeki ÇVÖA çalışmalarının performansı ile kıyaslanabilmesi için bu yöntem elde edilen veriler üzerinde denenmiştir. Sonuçlar, çalışma kapsamında ortaya çıkarılan YVA modellerin, bilişsel süreçleri etiketleme konusunda literatürdeki ÇVÖA yöntemlerinden daha başarılı olduğunu göstermektedir.

ÇVÖA yöntemlerini kısaca özetleyelim: Herhangi bir bilişsel süreç sırasında elde edilen etiketli fMRG verileri ön işlemeden geçirildikten sonra, bir sınıflandırıcı ile voksel yoğunluklarının bir makine öğrenme algoritması aracılığıyla öğretilerek tanınmasını sağlar. Literatürde bu amaç için yaygın olarak kullanılan algoritmalar, Yapay Sinir Ağları, Destek vektör Makinaları vb. popüler öğrenme ve tanıma yöntemleridir.

ÇVÖA literatüründe, detayları Bishop (2007)'de [22] açıklanan Destek Vektör Makineleri (DVM), k en yakın komşuluk (k-NN), sinir ağları (NN), ve naive bayes sınıflandırıcı yöntemleri yoğun olarak kullanılmaktadır. Projede geliştirilen matematiksel model ile literatürde yer alan diğer çalışmaları karşılaştırmak amacıyla, kullanılan bu sınıflandırıcılar toplanan deney verileri üzerinde denenmiştir. FMRG yönteminden gelen ham verinin sınıflandırıcılara vermek için uygun hale getirilmesinde detayları 3.2 numaralı başlıkta açıklanan görüntü işleme yöntemleri kullanılmıştır.

Toplanan verisetlerini etiketleme konusunda literatürde kullanılan sınıflandırıcılardan hangisinin daha başarılı olduğunu test etmek amacıyla 2. deney türüne ait veriler üzerinde 4 farklı sınıflandırıcı için performanslar elde edilmiştir. Tablo 1'den anlaşılacağı gibi, ortaya çıkan sonuçlar, DVM yönteminin projede elde edilen veriler için daha uygun olduğunu ortaya koymaktadır. DVM yönteminin başarılı olması elde edilen verilerdeki gürültünün sinyala oranı ve az sayıdaki örneklem miktarı düşünüldüğünde beklenen bir sonuçtur. DVM yönteminin 2. Deney türündeki başarısı gözönünde bulundurularak, oluşturulan matematiksel model ile alınan sonuçlarla literatürdeki sonuçlar karşılaştırılırken 1. Deney türü için de DVM yöntemi ana yöntem olarak kullanılmıştır. 1. Deney türü için elde edilen sonuçlar 3.4 numaralı bölümde detaylı olarak incelenmektedir.

Tablo 1 – 2. Deney türüne ait veriler üzerinde farklı sınıflandırıcı performansları.

	<b>DVM</b>	<b>k-NN</b>	<b>NN</b>	<b>NB</b>
<b>Denek 1</b>	<b>68</b>	60	10	55
<b>Denek 2</b>	<b>73</b>	57	34	51
<b>Denek3 3</b>	<b>74</b>	58	16	57
<b>Denek 4</b>	<b>75</b>	49	27	49
<b>Denek 5</b>	<b>69</b>	52	15	49
<b>Ortalama</b>	<b>71.8</b>	55.2	20.4	52.2

### 3.4. İnsan Beyninde Bağlanırlık

Klasik ÇVÖA yöntemindeki gibi voksel yoğunluk değerlerinin ardarda eklenerek bir öznitelik vektörü oluşturması bilişsel süreci modellemek için çok uygun değildir. Zira, voksellerden farklı zaman noktalarında elde edilen ölçümler tüm sınıflar için birbirine çok yakın değerler vermektedirler. Bu yaklaşım ile elde edilen öznitelik vektörleri sınıfların ayırt edilebilmesi için yeterli olamamaktadır. Tüm veri setlerinde yaptığımız deneylerde klasik ÇVÖA yöntemlerindeki performansların şans değerinden hafifçe yüksek olduğu gözlenmiştir. Kanımızca, mevcut yöntemlerin en önemli eksiklerinden birisi insan beynindeki **bağlanırlık** özelliğini göz önüne almamalarından kaynaklanmaktadır. Yerel voksel ağları yönteminin detaylarını açıklamadan önce insan beyninin çok önemli bir özelliği olan **bağlanırlık** kavramından bahsedelim:

Bağlanırlık, nöronlar arasındaki iletişimin düzeyini gösteren bir kavramdır. fMRG verileri söz konusu olduğunda nöron düzeyinde bir veri mevcut değildir. fMRG verilerinin en küçük birimi yaklaşık 1 mm<sup>3</sup> lük bir hacim kaplayan vokseldir. Bir vokselin içinde birkaç yüzbin nöron yer almaktadır. Herhangi bir bilişsel süreç sırasında voksel içindeki aktivasyon değeri yaklaşık 1 saniye aralıkla elde edilen zaman serilerinden oluşmaktadır. Proje kapsamında bağlanırlık kavramı vokseller arasındaki etkileşimi göstermek için kullanılmaktadır. fMRG verileri göz önüne alındığında, insan beynde üç şekilde gözlemlenmektedir:

- **Yapısal bağlanırlık:** Nörolojik ve deneysel bulgular, insan beyninin çeşitli bilişsel süreçleri yöneten anatomik bölgelere ayrılmış olduğunu göstermektedirler. Yapısal bağlanırlık, beyin anatomik bölgeleri arasındaki fiziksel bağların bulunması ile ortaya çıkan (beyaz kısmı oluşturan sinaptik yollar) fizyolojik ilişkiler için tanımlanmıştır.
- **İşlevsel bağlanırlık:** Birbirinden uzak ve aralarında fizyolojik yolak bulundurmayan beyin bölgeleri, belirli işlevsel süreçler için birlikte hareket edebilirler. Bu tür ilişkilere

işlevsel bağlanırlık adı verilir. İşlevsel bağlanırlık çoğunlukla, istatistiksel korelasyon yöntemleri kullanılarak ölçülmektedir.

- **Efektif bağlanırlık:** Farklı beyin bölgeleri arasında oluşan işlevsel bağlanırlık bazı sebep ve sonuç ilişkileri oluşturabilirler. Efektif bağlanırlık bu sebep-sonuç ilişkilerini ortaya çıkarmayı hedeflemektedir.

Yukarıda ana hatları ile belirtilen nörobilimsel bağlanırlık bulguları kullanılarak proje kapsamında insan beynindeki bilişsel süreçleri modelleyen yeni bir beyin ağı modeli önerilmiştir. Aşağıdaki bölümde, Yerel Voksel Ağları adını verdiğimiz bu model açıklanacaktır.

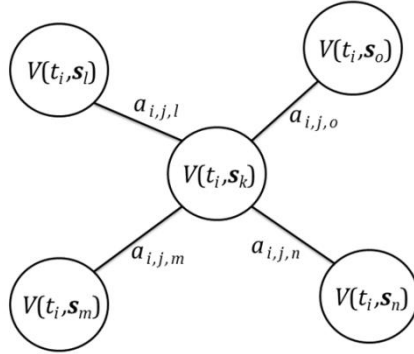
### 3.5. Yerel Voksel Ağları Yöntemi (YVA):

Projemiz kapsamında, yukarıda bahsedilen bağlanırlık kavramı vokseller arasındaki ilişkilerin modellenmesine indirgenmiştir. Yani, vokseller arası ilişkileri modelleyen yeni bir öznelik seti ve bu öznelik setini kullanan bir model önerilmiştir. **Yerel Voksel Ağları (YVA)** adını verdiğimiz bu model ile bilişsel süreçler vokseller arasındaki bağlanırlık cinsinden formüle edilmektedir. Bu maksatla, her bir bilişsel süreç, fMRG verilerinden elde edilen voksel ölçümleri arasındaki ağırlık vektörleri ile temsil edilir. Bu yöntemin özünde birbirine yakın olan vokseller arasındaki ilişkinin yerel olarak modellenmesi yatmaktadır. Bu ilişki doğrusal olarak modellenmektedir.

Önerilen bağlanırlık modelinde, her bir bilişsel süreç için elde edilen fMRG verileri (voksel yoğunluk değerleri) kullanılarak vokseller arasında yerel ağlar oluşturulur. Yerel ağların kenar ağırlıkları doğrusal kestirim yöntemleri ile hesaplanır. Oluşturulan yerel ağların kenar ağırlıkları öznelik vektörleri olarak birbirlerine eklenirler. Kenar ağırlıklarından oluşan öznelik vektörleri bir makina öğrenme algoritmasını eğitmek için kullanılırlar. Elde edilen kenar ağırlıkları vokseller arasındaki bağlanırlık derecesine karşı gelmektedirler.

Şimdi yukarıda kabaca açıkladığımız yerel voksel ağları modelin matematiksel olarak tanımlayalım: Öncelikle, en küçük hacim elemanı olan her bir vokseli  $v(t_i, \bar{s}_j)$ , ile ifade edelim. Bu gösterimde, voksel konumu  $\bar{s}_j$ , x, y ve z koordinatlarından oluşan üç boyutlu bir vektördür. Zaman parametresi olan  $t_i$  ise verilen uyarıcının ölçüldüğü anı göstermektedir. Önerilen Yerel Voksel Ağları, her bir voksel etrafında tanımlanmış olan yerel ağlardan oluşmaktadır. Yerel ağlar, merkez voksel etrafında tanımlanan p-komşuluk sistemi ile oluşturulmaktadır. Şekil 6'da bir  $t_i$  anında,  $\bar{s}_j$  konumunda bulunan bir voksel için oluşturulmuş olan örnek bir yerel voksel ağı sunulmaktadır. Yerel ağ üzerindeki  $a_{i,j,k}$ 'lar ağdaki kenar ağırlıkları olup (1) numaralı doğrusal bağlanım denkleminde göre hesaplanmaktadır.

$$v(t_i, \bar{s}_j) = \sum_{\bar{s}_k \in \eta_p} a_{i,j,k} v(t_i, \bar{s}_k) + \varepsilon_{i,j} \quad (1)$$



Şekil 6 - Voksel  $v(t_i, s_j)$  ve yerel komşuluğunu gösteren ağ diyagramı, en yakın 4 komşu ile oluşturulmuştur.

Bu denklemde,  $\varepsilon_{i,j}$ ,  $t_i$  anında voksel yoğunluk değeri  $v(t_i, \bar{s}_j)$  için hata payıdır. Kenar ağırlıkları,  $a_{i,j,k}$ , beklenen hatanın varyansı minimuma indirgenerek hesaplanmaktadır. Kenar ağırlıkları  $a_{i,j,k}$  merkez vokselin p-komşuluğundaki vokseller ile olan doğrusal ilişkisini gösterir. Her bir voksel için bir  $t_i$  anında hesaplanan bütün kenar ağırlıkları, beyinde  $t_i$  anındaki bilişsel sürecin **Yerel Voksel Ağları** ile ifade edilebilmesini sağlamaktadır. **Yerel Voksel Ağları** kenar ağırlıkları ile eğitilen makina öğrenme algoritmaları bilişsel sürecin bir makina öğrenme algoritmasıncı simule edilmesini hedeflemektedir ve örgü öğrenme algoritması olarak adlandırılmaktadır.

Kenar ağırlıkları  $a_{i,j,k}$  merkez vokselin p-komşuluğundaki vokseller ile olan doğrusal ilişkisini gösterir. Her bir voksel için bir  $t_i$  anında hesaplanan bütün kenar ağırlıkları, beyinde  $t_i$  anındaki bilişsel sürecin **Yerel Voksel Ağları** ile ifade edilebilmesini sağlamaktadır.

**Yerel Voksel Ağları**'nın kenar ağırlıkları ile eğitilen makina öğrenme algoritmaları bilişsel sürecin bir makina öğrenme algoritmasıncı simule edilmesini sağlamakta ve **Örgü Öğrenme Algoritması (Mesh Learning)** olarak adlandırılmaktadır.

### 3.5.1. Yerel Voksel Ağları Modelinde İşlevsel Bağlanırlık ve İşlevsel Komşuluk

Temel olarak, yerel voksel ağları kullanılarak geliştirilen örgü öğrenme yaklaşımı, bir bilişsel süreç için, her bir vokseli kendi yoğunluk derecesiyle değil, her bir zaman biriminde, o voksel için daha önceden belirlenmiş  $p$  adet uzamsal komşuluktaki voksel ile olan ilişkisi cinsinden ifade etmektedir. Bu ilişki doğrusal bir modelle ifade edilerek sınıflandırma problemleri çözülmektedir. Bağlanırlık bilgisi, son dönemlerde insan beyni çalışmalarında oldukça yoğun şekilde çalışılan bir bilgi olmakla birlikte, yerel örgü öğrenme yaklaşımın da geliştirilmesi için bir çok fırsat sunmaktadır. Bunlardan ilki, yerel örgülerin kurulduğu uzayın

tanımlanmasıdır. Örgü öğrenme yaklaşımı ilk olarak uzamsal olarak kurulmuş ve önerilmiştir. Ancak, bilişsel süreçler incelendiğinde, ayırım yapabilecek bilginin, her zaman bir voksel için uzamsal komşuluğunda olduğu kesin değildir. Bu maksatla, uzamsal olarak tanımlanmış komşuluk uzayının, işlevsel uzaya taşınması üzerinde durulmuştur. Yerel örgülerin doğrusallık varsayımı değiştirilmeden, sadece komşuluk tanımı değiştirilerek, yeni tanımlanan komşuluk uzayının bilişsel süreçleri daha gerçeğe yakın şekilde modelleyerek, sınıflandırma problemleri için daha fazla bilgi taşıyacak olan işlevsel komşuların seçimi üzerinde durmaktadır. Bu kapsamda yapılan çalışmalarımız, üç konferans (bir ulusal, iki uluslararası) ve bir dergi kenarını ile, ülkemiz ve dünya bilimine kazandırılmıştır.

Beynimizin yapısı göz önüne alındığında, birbirinden uzak nöronların aynı zihinsel süreçte aktif olabileceği gözlenmektedir. O halde, komşuluk kavramını uzamsal uzaklığa bağlı komşuluktan, işlevsel uzaklığa bağlı komşuluğa taşımamız daha geçerli bir yaklaşım olacaktır. Yani voksel koordinatları yakın olmasa da, benzer zaman serileri içeren vokselleri komşu sayabiliriz. İnsan beyninin karmaşık bir ağ şeklinde çalıştığı bilinmekle beraber, bilişsel sürece göre bu ağ yapısının değişimi, bilişsel bilimlerde oldukça yoğun çalışılan bir konudur (Rubinov ve Sporns, 2010 [23] ; Smith, Miller, Salimi, Webster, Beckmann, Nichols, Ramsey ve Woolrich, 2011 [24]). Bu alandaki çalışmalar, farklı bilişsel süreçlerde beynin ilintili bölgelerini bularak, bağlantı haritaları çıkarmakta ve kullanmaktadırlar. Bağlantı haritaları, her bir zihinsel süreç için beyin bölgeleri arasındaki ilinti oranını ifade eden bağlantı matrisleri ile ifade edilmektedirler. Bu matrisler, beynin ilgili alanındaki voksellerin, zaman serilerine göre birbirleriyle olan karşılıklı-ilinti, uyuşum, karşılıklı-entropi vb. fonksiyonları ile hesaplanmaktadırlar.

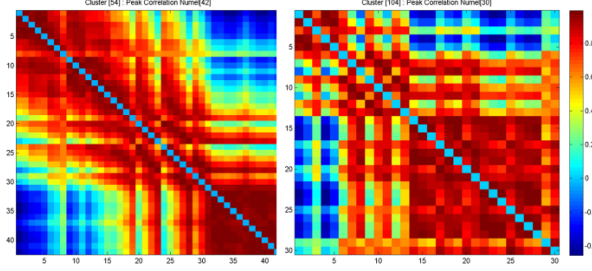
Bu projenin en önemli bilimsel yeniliği önerilen Yerel Voksel Ağları'nı oluştururken bağlantı haritalarını kullanmak ve böylelikle, her zihinsel işlem için beynimizin çeşitli bölgeleri arasındaki bağlantıları öğrenmektir. Bu maksatla projemizin ilk safhasında, yerel ağ komşuluk seçiminde uzamsal olarak en yakın komşu yerine işlevsel olarak en yakın komşu seçimi detaylı şekilde analiz edilmiştir. Yerel Voksel Ağları'nda bağlantı haritaları kullanılarak elde edilen ilk sınıflandırma performans sonuçlarında % 5-10 arasında iyileşme gözlemlenmektedir (Fırat, Özay, Önal, Öztekin, Vural, 2012 [25] ; Fırat, Özay, Önal, Öztekin, Vural, 2013 [16])

Matematiksel modelin oluşturulmasında, öncelikle beynin kısa süreli hafıza bölgesi, fiziksel konumuna göre ön tanımlı sayıda alt-kümeye bölünüp her bir alt-küme içerisindeki voksel yoğunluk değerlerinin zaman serileri incelenerek karşılıklı-ilinti matrisleri hesaplanacaktır. İki voksel  $v(.,\bar{s}_i)$  ve  $v(.,\bar{s}_j)$  zaman serileri arasındaki karşılıklı-ilinti Pearson korelasyon katsayısı  $\rho_{ij}$ , formül (2) ile bulunabilir.

$$\rho_{ij} = \frac{cov_{ij}}{\sqrt{var_i \cdot var_j}} \quad (2)$$



Denklem 2`de  $cov_{ij}$ ,  $i$  ve  $j$  vokselleri arasındaki kovaryansı,  $var_i$  ise ilgili vokselle için zaman içinde yoğunluk değışim varyansını belirtmektedir. Hesaplanan *karşılıklı-ilinti* matrisleri bilişsel süreçte benzer davranışlar sergileyen voksellerin ilişkisini vermekte olup şekil 7`de örnek iki farklı alt-küme için çıkarılan *karşılıklı-ilinti* matrisleri sunulmuştur.



Şekil 7 - İki farklı alt-küme için hesaplanan karşılıklı-ilinti matrisleri. Bu matrislerde +1`e yakın değerler (sıcak renkler) pozitif ilintiyi, -1`e yakın değerler (soğuk renkler) ise negatif ilintiyi ifade etmektedir.

Her vokselle  $v(t_i, \bar{s}_j)$  için p-komşuluk hesaplanırken, ilgili vokselin ait olduğu kümede kendisi ile en yüksek pozitif ilintide olan p-işlevsel-komşu seçilmiştir. İşlevsel komşu seçiminden sonra yerel ağlar işlevsel komşuluğa göre oluşturulmaktadır. Her bir vokselle etrafında oluşturulan bu Yerel Vokselle Ağ'ının kenar ağırlıkları  $a_{i,j,k}$ , denklem (1) deki hatanın varyansı en küçük kareler yöntemi ile minimize edilerek hesap edilmektedir. Kenar ağırlıklarının oluşturduğu vektör, o voksele ait özniteliği temsil etmektedir. Tüm voksellerin öznitelikleri kullanılarak bir öznitelik matrisi oluşturulmakta ve bu matris SVM, Yapay Sinir Ağları vb. makina öğrenme algoritmalarına girdi olarak verilerek sistem eğitilmektedir. Bilişsel sürecin sınıflandırılmasında karşılıklı-ilinti matrislerini kullanan ve böylece oluşturduğu yerel ağlarda vokseller arasındaki ilişkiyi doğrusal bağlantı modeli ile kestiren bu yöntemle İşlevsel Vokselle Ağları (İVA) adını veriyoruz.

### 3.5.2. Zaman Ölçümlü Vokselle Ağları Modelinde Uzamsal ve İşlevsel Komşuluk

Projede bahsedilen modellerdeki örgüler kurulurken en yakın komşular iki farklı şekilde seçilmiştir. Bunlardan ilki uzamsal komşuluk, ikincisi ise işlevsel komşuluktur.

Uzamsal komşuluk bulunurken voksellerin x,y,z koordinatlarına göre Öklid uzaklığı en küçük olan vokseller en yakın komşu olarak seçilir. Önce,  $l_k$  koordinatındaki vokselle için 1-en yakın komşuluk tanımlanır:

$$\eta_1[\bar{l}_j] = \{\bar{l}_k : \|\bar{l}_j - \bar{l}_k\| \leq \|\bar{l}_j - \bar{l}_o\|, \forall \bar{l}_o\}$$

Daha sonra tekrarlamalı olarak p-en yakın komşuluk belirlenir:

$$\eta_p^{spat}[\bar{l}_j] = \{\bar{l}_k \cup \eta_{p-1}[\bar{l}_j] : \|\bar{l}_j - \bar{l}_k\| \leq \|\bar{l}_j - \bar{l}_o\|, \forall \bar{l}_o \in \eta_{p-1}^{spat}[\bar{l}_j]^c\}$$

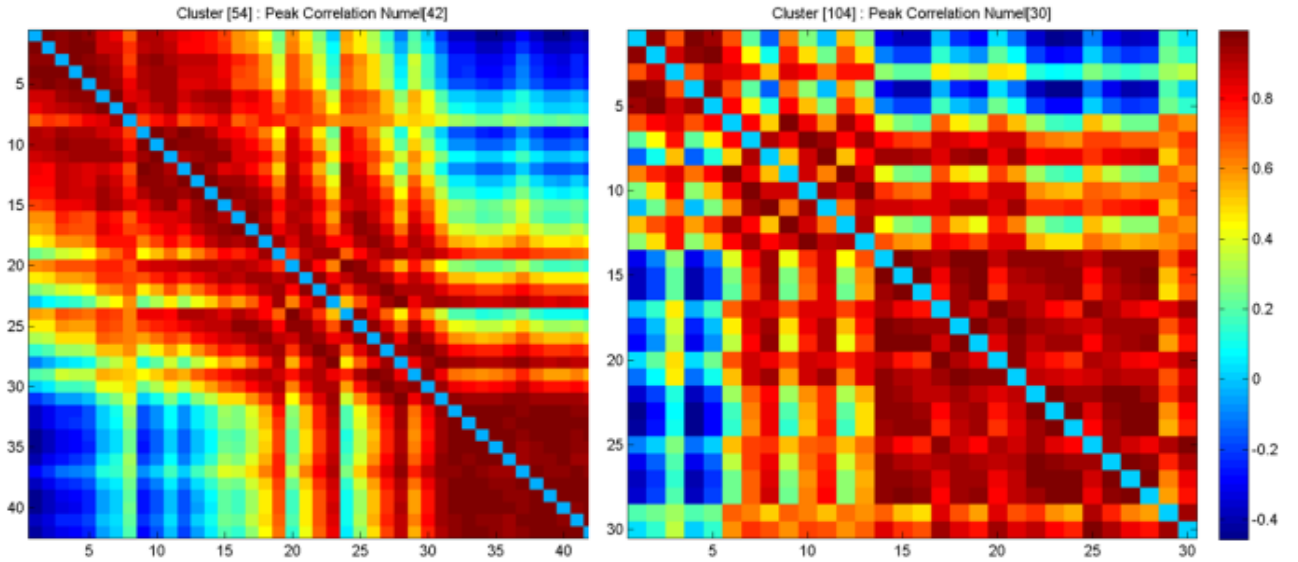
İşlevsel komşuluk ise voksellerin tepkileri arasında 3 farklı metot kullanılarak belirlenir:

- Pearson korrelasyonu
- Cosine benzerliği
- Öklid uzaklığı

Tüm deney süresince bir vokselden alınan tepkiler art arda eklenerek her bir vokselden aşağıdaki vektör elde edilebilir:

$$R(\bar{l}_j) = [\bar{r}(s_1, \bar{l}_j), \bar{r}(s_2, \bar{l}_j), \dots, \bar{r}(s_{N^{tr}}, \bar{l}_j)]$$

Daha sonra, bu vektörler arasında yukarıda bahsedilen 3 metot ile benzerlikler hesaplanabilir. Bu benzerlikler hesaplanırken  $X$  ve  $Y$  iki farklı vokselden deney süresince alınan tepkilerin vektörlerini temsil eder. Verisetindeki tüm vokal ikilileri arasında benzerlikler hesaplanarak bir bağlanırlık matrisi elde edilir. Her vokal için en yakın  $p$ -komşuları, o vokselin verilerinin ait olduğu satırdaki en yüksek benzerlik değerine sahip olan vokseller olarak belirlenebilir. Aşağıdaki benzerlik matrisinde kırmızı renkler pozitif benzerliği, mavi renkler negatif benzerliği temsil etmektedir. Koyu renkler yüksek mutlak benzerlik değerlerine, açık renkler düşük mutlak benzerlik değerlerine karşı gelmektedir.



## Pearson Korrelasyonu

Vektörler arası Pearson korrelasyonu matematiksel olarak aşağıdaki gibi hesaplanır:

$$cor(X, Y) = \frac{cov(X, Y)}{\sigma(X)\sigma(Y)}$$

Diğer bir deyişle, iki vektör arası kovaryanslar hesaplanır ve vektörlerin standart sapmalarının çarpımına bölünür. Böylelikle -1 ile 1 arasında bir değer elde edilir. Yüksek pozitif korrelasyon değeri voksellerin uyarılar karşısında benzer davrandığı anlamına gelir. Benzer şekilde yüksek negatif korrelasyon değeri voksellerin uyarılar karşısında zıt davrandıkları anlamına gelir. 0'a yakın korrelasyon değerleri ise vokseller arasında verilen tepkiler açısından bir ilişki olmadığını gösterir. Bizim çalışmalarımızda tüm vokseller arası korrelasyonlar hesaplanmış, her voksel için en yakın komşular korrelasyon değeri 1'e en yakın olanlar olarak seçilmiştir.

### **Cosine Benzerliği**

Vektörler arası Cosine benzerliği matematiksel olarak aşağıdaki gibi hesaplanır:

$$cos(X, Y) = \frac{XY^T}{(XX^T)(YY^T)}$$

Bu formül kullanılarak iki vektörün iç çarpımları hesaplanarak, her vektörün kendisiyle iç çarpımlarına bölünür. Elde edilen değer ne kadar büyükse, iki vektör arasındaki açı o kadar küçüktür ve vektörler birbirine o kadar benzerdir. Bu nedenle bizim metodlarımızda vektörler arası Cosine benzerliği diğer vektörlere göre en yüksek olan voksel en yakın komşu olarak seçilmektedir. Bir vokselin p-en yakın işlevsel komşusunu Cosine benzerliği kullanarak bulurken, o vokselin diğer tüm voksellerle arasındaki Cosine benzerliği hesaplanır ve maksimum benzerliğe sahip vokseller en yakın komşular olarak seçilir.

### **Öklid uzaklığı**

Vektörler arası Öklid uzaklığı matematiksel olarak aşağıdaki gibi hesaplanır:

$$euc(X, Y) = \sqrt{(X - Y)(X - Y)^T}$$

Bu formül kullanılarak iki vektörün tüm elemanları karşılıklı olarak birbirinden çıkarılır ve elde edilen farkın iç çarpımları hesaplanarak karekökü alınır. İki vektörün elemanları birbirine ne kadar yakınsa vektörler o kadar yakındır. Bu nedenle voksellerin tepkileri arasındaki en düşük Öklid uzaklığı en yüksek benzerlik anlamına gelir. Bir vokselin p-en yakın işlevsel komşusunu Öklid uzaklığı kullanarak bulurken, o vokselin diğer tüm voksellerle

arasındaki Öklid uzaklığı hesaplanır ve minimum uzaklığa sahip vokseller en yakın komşular olarak seçilir.

Hem uzamsal hem de işlevsel komşuluk bulunurken Öklid uzaklığı kullanılmıştır. Ancak, dikkat edilecek nokta şudur ki, uzamsal komşuluk hesaplanırken voksellerin x,y,z koordinatları arası Öklit uzaklığa bakılırken işlevsel komşuluk hesaplanırken voksellerden gelen tepkiler arası Öklit uzaklığa bakılmıştır.

### 3.5.3. Zaman Ölçümlü Yerel ve İşlevsel Voksel Ağları

Projemizde, daha önce yine projemiz kapsamında geliştirilen Yerel Voksel Ağları ve İşlevsel Voksel Ağları modelleri, vokseller tarafından bir uyarana gösterilen süreç göz önünde bulundurularak zamansal ölçümler ile zenginleştirilmiş ve “Zaman ölçümlü Yerel ve İşlevsel Voksel Ağları” modelleri önerilmiştir. Bu kapsamda yapılan çalışmamız iki uluslararası konferans kenarını ile ülkemiz ve dünya bilimine kazandırılmıştır.

Bir uyarana karşısında aktif olan bölgelerdeki nöronlar mevcut oksijeni tüketir ve aktif bölgelerde oksijen ihtiyacı doğar. Bu ihtiyacı gidermek için oksijenli kan aktif bölgelere hücum eder. Sonrasında nöronlar yeni gelen oksijeni tüketir ve hemodinamik tepki denge değerine gelir. Görüldüğü gibi, aktif bölgede önce oksijendeki kan miktarı artar, 6 saniye civarında maksimum değere ulaşır ve 12 saniye civarında da denge haline geri döner. fMRG deneyleri esnasında her 2 saniyede bir ölçüm alındığını göz önünde bulundurulursa, her vokselde bir uyarana karşı gösterilen her bir tepki için 6 ölçüm alınmaktadır. Daha önceki çalışmalarda bu 6 ölçümün tepe noktası olan 3. Ölçüm değerleri kullanılmış ve diğer 5 ölçüm göz ardı edilmiştir. Önerilen modelimizde bu 6 ölçümün tamamı kullanılarak modeller geliştirilmiştir. Bu modele göre bir vokselde alınan tepki aşağıdaki gibi tanımlanmıştır:

$$\bar{r}(s_i, \bar{l}_j) = [v(s_i, t_1, \bar{l}_j), v(s_i, t_2, \bar{l}_j), \dots, v(s_i, t_W, \bar{l}_j)]^T$$

Diğer bir deyişle, voksellerin uyarana gösterdiği her bir tepki 6 voksel yoğunluk değerinden oluşmaktadır. Her bir voksel etrafında en yakın komşuları ile aşağıdaki gibi bir örgü kurulur.

$$\bar{r}(s_i, \bar{l}_j) = \sum_{\bar{l}_b \in \eta_p} a_{i,j,b} \bar{r}(s_i, \bar{l}_b) + \bar{\epsilon}_{i,j}$$

Bu denklemdeki  $\eta_p$  sembolü p-en yakın komşu voksellerin oluşturduğu kümeyi göstermektedir. Yani;  $\eta_p$  setinin içinde her bir örgüdeki  $v(t_i, \bar{s}_j)$  merkez voksele en yakın p adet voksel bulunmaktadır. Komşuluk fiziksel ya da işlevsel olabilir. Her iki durumda da aynı denklem kullanılarak örgü kenar ağırlıkları hesaplanır. Bu denklemde yalnızca komşuluğa göre merkez voksel etrafındaki p-en yakın vokseller değişecektir. Her bir sınıfı temsil eden uyarı için

6 voksel yoğunluk değeri ölçüldüğü göz önünde bulundurulduğunda yukarıdaki denklem açık olarak aşağıdaki gibi yazılabilir:

$$\begin{pmatrix} v(s_i, t_1, \bar{l}_j) \\ v(s_i, t_2, \bar{l}_j) \\ \vdots \\ v(s_i, t_W, \bar{l}_j) \end{pmatrix} = \sum_{\bar{l}_b \in \eta_p} a_{i,j,b} \begin{pmatrix} v(s_i, t_1, \bar{l}_b) \\ v(s_i, t_2, \bar{l}_b) \\ \vdots \\ v(s_i, t_W, \bar{l}_b) \end{pmatrix} + \begin{pmatrix} \varepsilon_{i,1,j} \\ \varepsilon_{i,2,j} \\ \vdots \\ \varepsilon_{i,W,j} \end{pmatrix} .$$

Örgünün düğümleri voksellerden alınan zaman serilerinin değerlerini göstermektedir. Merkezdeki voksel ile en yakın p-komşu voksel arasındaki kenarların ağırlıkları ridge regresyon yöntemi kullanılarak, hatanın karesinin beklenti değeri minimize edilecek şekilde aşağıdaki gibi kestirilir ve voksellerde her bir sınıf için ölçülen yoğunluk değerleri komşu voksellerin yoğunluklarının lineer kombinasyonu cinsinden ifade edilir. Böylece hatanın karesinin beklenti fonksiyonu, aşağıdaki formülün en iyilenmesi ile elde edilir;

$$E \left( (\bar{\varepsilon}_{i,j})^2 \right) = E \left( \left( \bar{r}(s_i, \bar{l}_j) - \sum_{\bar{l}_b \in \eta_p} a_{i,j,b} \bar{r}(s_i, \bar{l}_b) \right)^2 \right) .$$

Yukarıdaki denklemin sağ tarafının elde etmek istediğimiz kenar ağırlıkları olan  $a_{i,j,b}$  ye göre türevi alınarak sifıra eşitlenirse, kapalı formda ridge regresyonunun çözümü aşağıdaki formülle gösterilebilir:

$$\bar{a}_{i,j} = (Q_{i,j}^T Q_{i,j} + \lambda I)^{-1} Q_{i,j}^T \bar{r}(s_i, \bar{l}_j) .$$

Yukarıdaki formüldeki Q değişkeni, komşu voksellerden elde edilen zaman serilerinin birleşimiyle oluşan matrislerdir ve aşağıdaki gibi gösterilir:

$$Q_{i,j} = [\bar{r}(s_i, \bar{l}_b)], \forall \bar{l}_b \in \eta_p$$

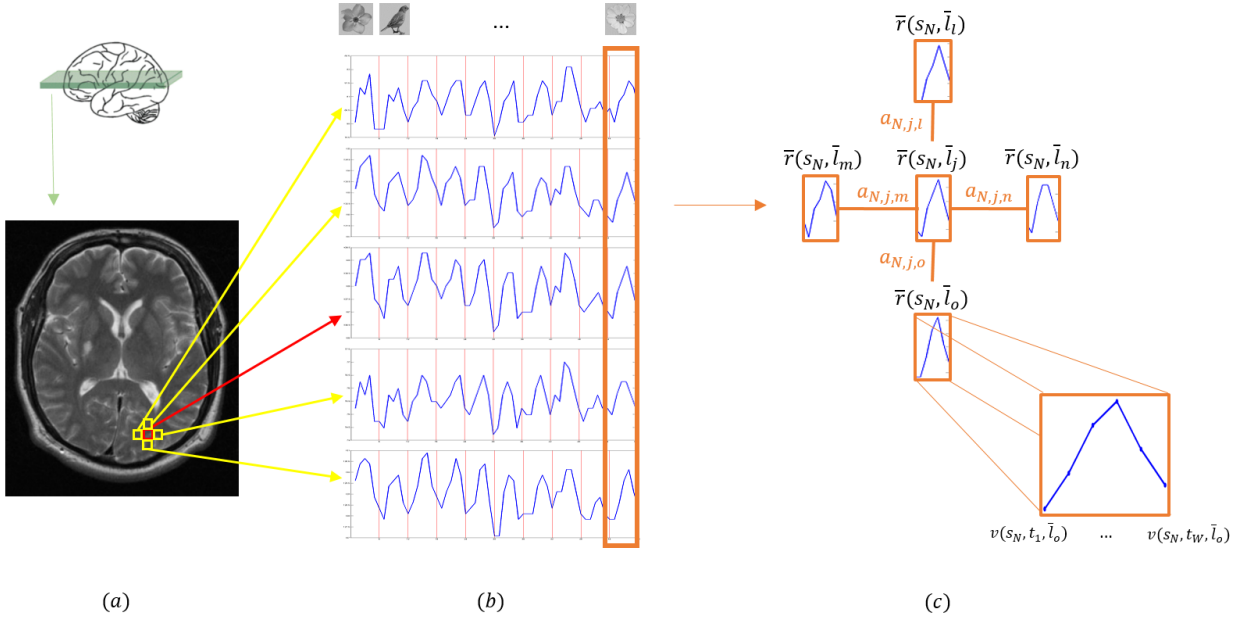
Her bir merkez voksel etrafında oluşturulan Yerel Voksel Ağının kenar ağırlıklarının kestirilmesi ile elde edilen Yerel Voksel ağları tüm beyin için bir beyin ağı oluşturur. Böylece, fMRG ölçümler sırasında elde edilen voksel zaman serileri yerine, her bir uyarı için kenar ağırlıkları değişen, ancak, topolojik yapısı değişmeyen bir beyin ağı elde edilmiş olur.

#### 3.5.4. Zaman Ölçümlü Yerel Voksel Ağları (ZÖYVA)

Bir önceki bölümde matematiksel tanımı yapılan **YVA** yönteminde, vokseller arası komşuluk, voksellerin koordinatları arasındaki öklit uzaklığına göre belirlenmekte ve merkez voksele en yakın komşu olan vokseller seçilerek yerel örgüler kurulmaktadır (**ZÖYVA**).

Şekil 8'de bir beyin kesiti görülmektedir. Örnek olarak, alınan kırmızı renkli bir vokseli merkez voksel olarak kabul edelim. Etrafındaki sarı renkle gösterilen 4 voksel bu örnek için merkez vokselin 4-en yakın komşusudur. Deney süresince gösterilen her bir uyaran için tüm

voksellerden gelen tepkiler ölçülmektedir. Son örnekte gösterilen uyarı için 6 vöksel yoğunluk deęeri ieren zaman serileri turuncu kutular iinde gsterilmiřtir. rnekteki merkez vokselde ölçülen zaman serisi ile p-en yakın yerel komřularından gelen zaman serileri kullanarak bir örgü kurulur. Bu örgünün kenar aęırlıkları ridge regresyon kullanarak kestirilir ve merkez vöksel bu aęırlıklar cinsinden ifade edilir. Kestirim sırasında, verilen bir uyarı için elde edilen 6 adet yoğunluk deęeri olan zaman serisi kullanıldıęı iin mevcut metotlardan farklı olarak hemodinamik tepkideki (Hemo Dynamic Response) tüm zamansal ölçümler de modellemede kullanılmaktadır. Dięer bir deyiřle, her bir örgünün kenar aęırlıkları, merkez vokselin p-en yakın komřuları ile arasındaki iliřkiyi gstermektedir.



Şekil 8 - Zaman Ölçümlü Yerel Vöksel Aęları Modeli. Merkez vokselin 4-en yakın fiziksel komřusu belirlenir (a). Tüm uyarılara karřılık verilen vöksel tepkileri deney süresince ölçülür (b). rnek bir uyarıya verilen tepkiler kullanarak merkez vöksel ve p-en yakın komřuları ile bir örgü kurulur (c). Bu örgünün kestirilen kenar aęırlıkları merkez vokseli ifade etmede kullanılır.

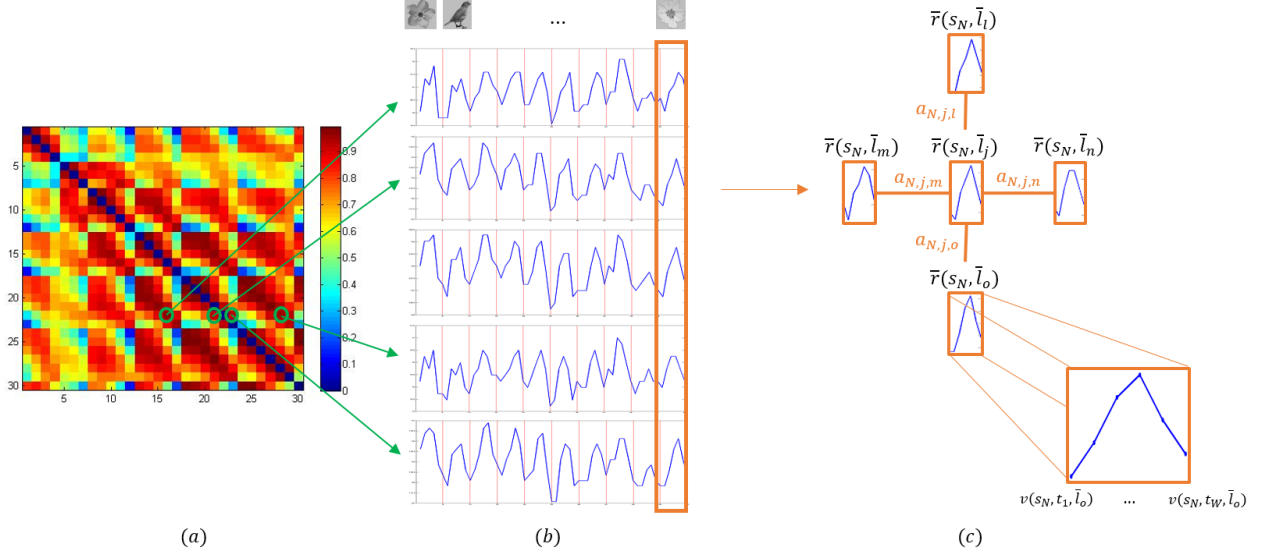
### 3.5.5. Zaman Ölçümlü İşlevsel Vöksel Aęları (ZÖİVA)

Bu yöntemde vokseller arası komřuluk, voksellerden ölçülen zaman serileri arasında ařaęıdaki ölçüm kriterleri kullanılmıřtır

1. Pearson korrelasyonu (**ZÖİVA-cor**),
2. Cosine benzerlięi (**ZÖİVA-cos**) ve
3. Öklit uzaklıęı (**ZÖİVA-ök**)

Bu kriterler kullanarak tanımlanan p- en yakın komřuluk sistemleri belirlenmiř ve her bir vöksel etrafında p-en yakın komřu vöksel seti seilerek yıldız topolojisi ile işlevsel örgüler kurulmuřtur.

Öncelikle, her 3 kriter için de tüm vokselle ikilileri arası benzerlikler hesaplanarak Şekil 9'daki gibi işlevsel benzerlik matrisleri oluşturulur. Daha sonra, yine merkez vokselle ve p-en yakın işlevsel komşularından gelen zaman serileri tüm deney süresince kaydedilir. Merkez vokselle etrafında işlevsel komşuları ile bir örgü kurulur. Kurulan işlevsel örgünün kenar ağırlıkları merkez vokselle ile işlevsel komşuları arasındaki ilişkiyi temsil eder. Benzer şekilde kenar ağırlıkları kestirilirken hemodinamik tepkinin yalnızca tepe noktasındaki değeri değil, tüm ölçümler kullanılmıştır.



Şekil 9 - Zaman Ölçümlü İşlevsel Vokselle Ağları Modeli. Merkez voksellein 4-en yakın fiziksel komşusu belirlenir (a). Tüm uyarılara karşılık verilen vokselle tepkileri deney süresince ölçülür (b). Örnek bir uyarıya verilen tepkiler kullanılarak merkez vokselle ve p-en yakın komşuları ile bir örgü kurulur (c). Bu örgünün kestirilen kenar ağırlıkları merkez vokselle ifade etmede kullanılır.

### 3.6. Geliştirilen Yöntemlerin Karşılaştırılması

Bu projede öncelikle zamansal bilgiyi içermeyen Yerel Vokselle Ağları (YVA) ve İşlevsel Vokselle Ağları (İVA) metotları geliştirilmiştir. Sonrasında Zaman Ölçümlü Yerel Vokselle Ağları (ZÖYVA) metodu ile farklı benzerlik ölçümleri kullanan Zaman Ölçümlü İşlevsel Vokselle Ağları (ZÖİVA-cor, ZÖİVA-cos, ZÖİVA-ök) metotları geliştirilmiştir. Geliştirdiğimiz yöntemlerin sınıflandırma performanslarının başarısını anlamak için bu yöntemlere ek olarak literatürdeki mevcut yöntemler de sınıflandırma sonuçlarının elde edilmesinde kullanılmıştır. Bu amaçla proje kapsamında geliştirilen metotlar, aşağıdaki metotlar ile karşılaştırılmıştır:

**Çoklu Vokselle Örüntü Analizi (ÇVÖA):** Bu yöntemde her vokselden elde edilen vokselle yoğunluk değeri art arda eklenerek öznitelik vektörü oluşturulur. Bu vektörler kullanılarak sınıflandırıcılar eğitilir ve test edilir. Öznitelik vektörü çıkarılırken 6 ölçümlük tepkinin 3. ölçümü kullanılarak hemodinamik tepki fonksiyonunun tepe noktası (ÇVÖA-maks) ya da tüm hemodinamik tepki fonksiyonunun ortalaması (ÇVÖA-ort) alınmış ve her iki öznitelik de kullanılarak sınıflandırma sonuçları elde edilmiştir.

**Fonksiyonel Bağlılık (FB):** Bu yöntemde vokseller arası ikili ilişkiler öznitelik olarak kullanılmaktadır. Her vokselin fiziksel olarak p-en yakın komşuları ile ikili uzaklıkları, Pearson korelasyonu (**FB-cor**), Cosine benzerliği (**FB-cos**) ve Öklit uzaklığı (**FB-ök**) yöntemleri kullanılarak çıkarılmıştır. Daha sonra, her vokselin kendi komşuluğundaki ikili uzaklıkları art arda eklenerek öznitelikler elde edilir.

### 3.6.1. Sınıflandırma Sonuçları

Çalışmamızda 3 çeşit öznitelik vektör uzayı oluşturulmuş ve bunlar kullanılarak k-en yakın komşu (k-NN), Destek Vektör Makinesi (DVM), Yapay Sinir Ağları (NN) ve Naive Bayes (NB) sınıflandırıcıları eğitilmiş ve test edilmiştir. Elde edilen öznitelik vektörleri ham data, yerel örgü ağları ve zaman ölçümlü yerel örgü ağlarıdır. Yalnızca nesne tanıma deneyinden elde edilen ham data üzerinde tüm bu 4 sınıflandırıcı performansları elde edilmiş, sonuç olarak ham data üzerinde en yüksek sonuç veren Destek Vektör Makinesi sınıflandırıcısıyla geri kalan deneylere devam edilmiştir. Ham data üzerinde 4 sınıflandırıcı ile elde edilen sınıflandırma performansları aşağıdaki tabloda görülebilir:

*Tablo 2 - 2. Deney türünden elde edilen veriler için ham veri kullanılarak eğitilen ve test edilen 4 ayrı sınıflandırıcı performansları.*

	<b>DVM</b>	<b>k-NN</b>	<b>NN</b>	<b>NB</b>
<b>Denek 1</b>	<b>68</b>	60	10	55
<b>Denek 2</b>	<b>73</b>	57	34	51
<b>Denek 3</b>	<b>74</b>	58	16	57
<b>Denek 4</b>	<b>75</b>	49	27	49
<b>Denek 5</b>	<b>69</b>	52	15	49
<b>Ortalama</b>	<b>71.8</b>	55.2	20.4	52.2

Proje kapsamında geliştirilen metotlar ve literatürdeki metotlar kullanılarak nesne tanıma veriseti üzerinde Tablo 3'deki DVM sınıflandırma sonuçları elde edilmiştir. Zaman Ölçümlü İşlevsel Voksel ağları üç farklı benzerlik metriği kullanılarak (**ZÖİVA-cor**, **ZÖİVA-cos**, **ZÖİVA-ök**) çalıştırılmıştır. Zamansal ölçüm içermeyen İşlevsel Voksel Ağları ve Yerel Voksel Ağları yöntemlerinde ağlar kurulurken voksel tepkilerinin tepe noktaları (**İVA-maks**, **YVA-maks**) ve tepkilerin ortalamaları (**İVA-ort**, **YVA-ort**) kullanılmıştır. Benzer şekilde literatürdeki fonksiyonel benzerlik metodu kullanılırken üç farklı metrik (**FB-cor**, **FB-cos**, **FB-ök**) kullanılmıştır.



Tablo 3– 2. Deney türünden elde edilen veriler için DVM kullanılarak elde edilen (%) sınıflandırma performansları.

	Denek 1	Denek 2	Denek 3	Denek 4	Denek 5	Ortalama
<b>ZÖİVA-cor</b>	90	88	88	82	78	<b>84.9</b>
<b>ZÖİVA-cos</b>	90	85	90	82	75	<b>84.3</b>
<b>ZÖİVA-ök</b>	93	89	86	81	79	<b>85.6</b>
<b>ZÖYVA</b>	87	88	81	83	75	<b>82.6</b>
<b>İVA-ort</b>	80	78	81	78	75	<b>78.2</b>
<b>İVA-maks</b>	75	69	88	83	78	<b>78.6</b>
<b>YVA-ort</b>	82	72	74	74	74	<b>74.9</b>
<b>YVA-maks</b>	75	74	78	75	76	<b>75.6</b>
<b>FB-cor</b>	65	74	61	58	50	<b>61.6</b>
<b>FB-cos</b>	50	51	51	50	49	<b>50.2</b>
<b>FB-ök</b>	72	69	81	83	71	<b>75.1</b>
<b>ÇVÖA-ort</b>	70	71	72	78	69	<b>72</b>
<b>ÇVÖA-maks</b>	68	67	69	83	72	<b>72</b>

Tablo 3'den görülebileceği gibi, en yüksek ortalama performans zaman ölçümlü işlevsel voksel ağları metodunda Öklid uzaklığı kullanıldığında (**ZÖİVA-ök**) elde edilmiştir. Ek olarak, korrelasyon ve cosine benzerliği kullanıldığında da (**ZÖİVA-cor**, **ZÖİVA-cos**) diğer metotlara göre daha yüksek performanslar elde edilmiştir. Bu performansları zaman ölçümlü yerel voksel ağları metodu takip etmektedir. Sonuçlar göstermektedir ki, ağ kurulurken zamansal ölçümlerin göz önünde bulundurulması, yalnızca tek bir voksel yoğunluk değerinin kullanılmasına göre daha yüksek performans vermektedir. Diğer bir deyişle, ZÖİVA ve ZÖYVA metotları, İVA ve YVA metotlarından daha başarılıdır. Vokseller arası ikili ilişkilerde cosine benzerliği (FB-cos) ve korrelasyon (FB-cor) kullanımı en düşük performanslara yol açmıştır. İkili ilişkilerde (FB-ök) diğerlerine kıyasla daha iyi performansa sahip olmasına rağmen geliştirdiğimiz yöntemlerden daha düşük performans vermektedir.

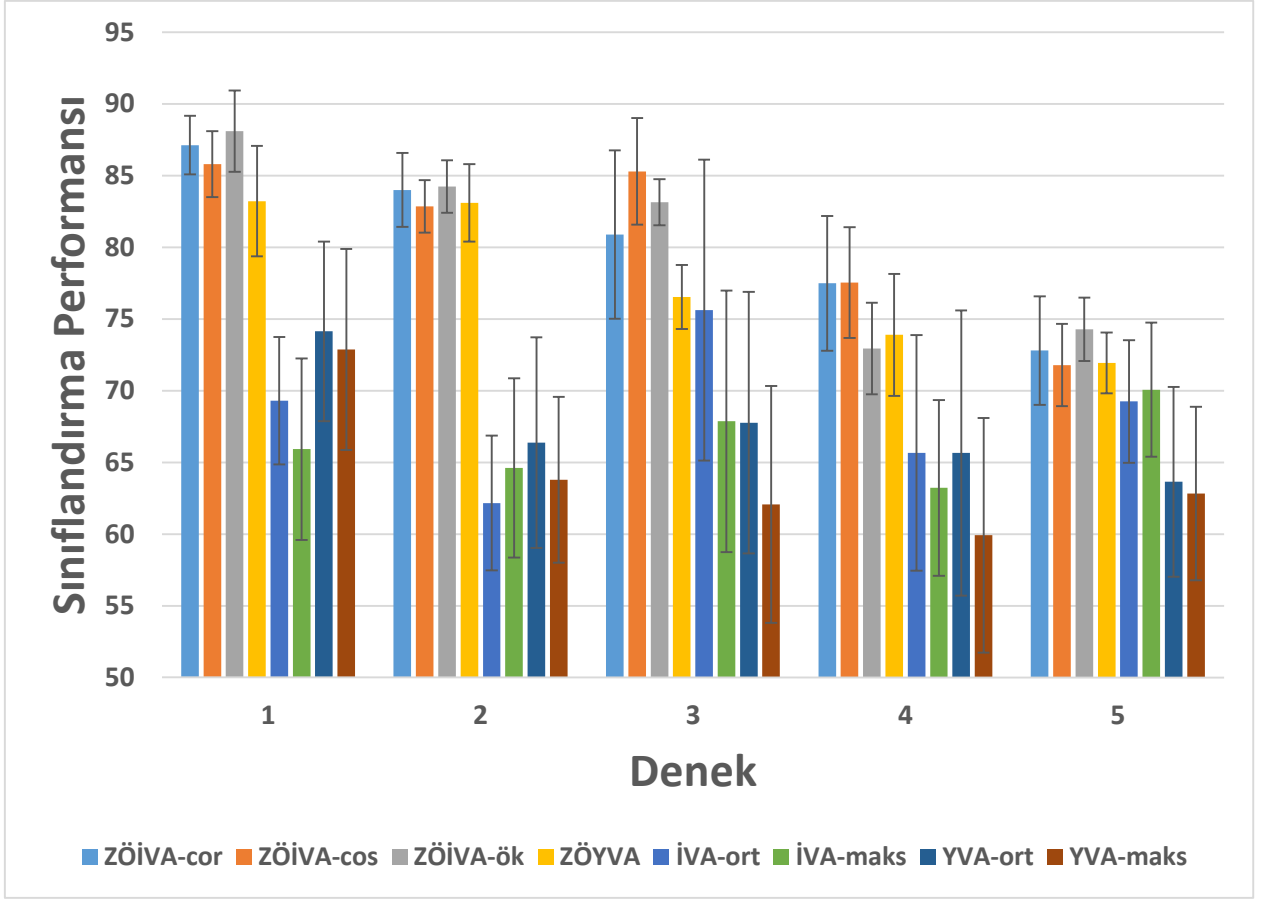
Deney:1 için elde edilen verilerde, her bir uyarıdan sonra yeterince süre verilmediği gözlemlenmiştir. Bu durum, projemizin ilk aşamalarında geliştirilmiş olan ve zamansal ölçüm içermeyen **Yerel Voksel Ağları (YVA)** yöntemi için bir problem teşkil etmemektedir. Ancak sonrasında geliştirilen ve zamansal ölçümü, yani vokseller tarafından gösterilen hemodinamik tepkinin tümünü göz önünde bulundurulan yöntemler, hafıza verisetine uygun değildir. Buna rağmen ZÖYVA yöntemi ile elde edilen sonuçlar YVA metotunun ve literatürdeki diğer yöntemlerin sonuçlarından daha iyi çalışmıştır. Sonuçlar Tablo 4'te aşağıdaki gibi görülebilir:

Tablo 4 – 2. Deney türünden elde edilen veriler için DVM kullanılarak elde edilen (%) sınıflandırma performansları

	<b>ZÖYVA</b>	<b>YVA</b>	<b>FB-cor</b>	<b>FB-cos</b>	<b>FB-euc</b>	<b>ÇVÖA</b>
<b>Denek 1</b>	68	51	65	63	64	39
<b>Denek 2</b>	68	65	68	68	68	54
<b>Denek 3</b>	66	60	64	62	63	46
<b>Denek 4</b>	67	64	62	60	60	47
<b>Denek 5</b>	66	67	61	62	63	49
<b>Denek 6</b>	68	58	68	64	67	47
<b>Denek 7</b>	67	56	62	61	61	44
<b>Denek 8</b>	66	63	63	60	61	52
<b>Denek 9</b>	67	55	64	64	64	43
<b>Ortalama</b>	<b>67</b>	<b>60</b>	<b>64</b>	<b>62</b>	<b>63</b>	<b>47</b>

### 3.6.2. Her Bir Yerel Voksel Vğındaki p- En Yakın Komşu Sayısının Belirlenmesi

Model derecesi, her vokselin kaç adet komşu voksel ile ifade edileceği anlamına gelmektedir. Proje kapsamında yapılan ilk çalışmalarda voksel ağındaki komşu sayısı, yani model derecesi bilgi teoretik kriterler kullanılarak kestirilmeye çalışılmıştır. Ancak verisetlerinin yapısı gereği sınıflandırma performansları model derecesine göre ciddi değişiklikler göstermemektedir. Bu durum aşağıdaki figürde model derecesine (p'ye) göre sınıflandırma performanslarının değişimiyle görülebilir:



Bu figürde bar yükseklikleri belirli bir aralıkta (  $p = [2,3,...30]$  ) komşu sayıları için elde edilen ortalama performans değerlerini göstermekte olup, hata barı yükseklikleri sınıflandırma performanslarının standart sapmasını göstermektedir. Standart sapmanın düşüklüğü, sınıflandırma performanslarının  $p$ 'ye göre çok değişmediğini göstermektedir. Figürden görülebileceği gibi ZÖİVA-cor, ZÖİVA-cos, ZÖİVA-ök ve ZÖYVA metotları için standart sapmalar oldukça düşüktür. Zamansal ölçümleri kullanan yöntemlere kıyasla zamansal ölçümleri göz önünde bulundurmayan yöntemlerde standart sapma daha yüksektir ancak performanslar düşünüldüğünde bu fark ciddi bir fark değildir. Bu nedenle komşu sayısı bilgi teoretik yöntemlerle kestirilme yerine, en yüksek sınıflandırma performansı veren komşuluk değeri ideal komşu sayısı olarak seçilmiştir.

### 3.7. Algoritmaların Paralleleştirilmesi ve Optimizasyonu

Proje kapsamında, yerel voksel ağları çıkarımı gösterilen her bir örnek için ve ilgi bölgesindeki her bir voksel için yapılmaktadır. Verisetinde yaklaşık 2500 voksel ve yaklaşık 200 örnek olduğu düşünülürse 2500x200 kez vokseller etrafında komşuları ile kenar ağırlıkları kestirilme işlemi tekrarlanmaktadır. Ek olarak, komşuluk değeri değiştikçe, kurulan her bir örgü ağı için kenar ağırlıkları kestirim süresi uzamaktadır. Örneğin; 20 en yakın komşu ile kurulan örgü kenar ağırlıklarının hesaplanması, 5 en yakın komşu ile kurulan örgü kenar ağırlıklarının hesaplanmasından daha uzun sürmektedir. Örgülerin seri olarak kurulması oldukça zaman

almaktadır, ve her bir örgü diğerinin bilgisinden bağımsız çalışabildiği için örgü kenar ağırlıklarının kestimi paralelleştirmeye oldukça uygundur. Bu projede sınıflandırma için kullanılacak örgü kenar ağırlıklarının elde edilme süreci CUDA kullanılarak GPU üzerinden paralelleştirilmiştir. Parallelleştirme yapabilmek için süreç aşağıdaki adımlara bölünmüştür:

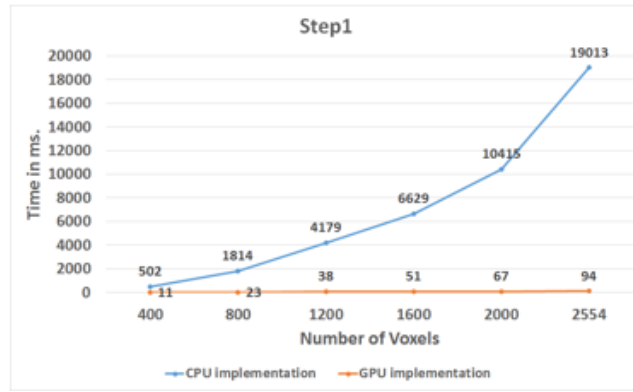
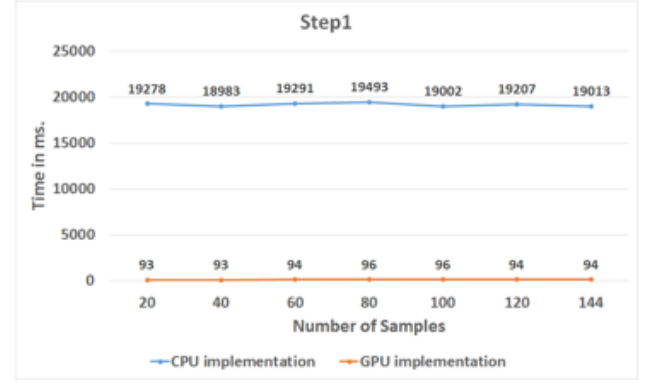
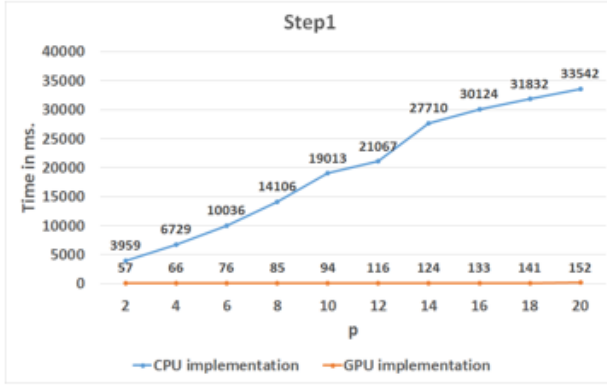
1. Verinin (voksellerden gelen tepkilerin) 2 boyutlu matrise okunması
2. Voksel xyz koordinatlarının 2 boyutlu matrise okunması
3. Voksellerin p-en yakın komşuluklarının bulunması
  - a. Merkez voksel ile diğer tüm vokseller arasında ikili Öklid uzaklıklarının hesaplanması
  - b. Voksel indislerinin merkez voksele olan uzaklıklarına göre sıralanması
  - c. En yakın p-vokselin seçilmesi
  - d. 3.a – 3.c arası adımların verisetindeki tüm vokseller için yapılması ve uzaklık matrisinin elde edilmesi
4.  $\bar{a}_{i,j} = (Q_{i,j}^T Q_{i,j} + \lambda I)^{-1} Q_{i,j}^T \bar{r}(s_i, \bar{l}_j)$  denkleminin  $\bar{l}_j$  konumundaki tüm vokseller ve tüm örnekler  $s_i$  için çözülmesi
  - a.  $Q_{i,j}$  matrisinin 3. Adımda bulunan p-en yakın komşulardan gelen voksel tepkileri kullanılarak oluşturulması
  - b.  $Q_{i,j}$  matrisinin transpozunun alınması (matris transpozu)
  - c.  $Q_{i,j}^T$  matrisinin  $Q_{i,j}$  matrisi ile çarpımı (matris – matris çarpımı)
  - d.  $\lambda I$  matrisinin 4.3 adımındaki matrise eklenmesi (köşegendeki elemanlara  $\lambda$  eklenmesi)
  - e.  $(Q_{i,j}^T Q_{i,j} + \lambda I)^{-1}$  hesaplanması (matris tersi)
  - f.  $(Q_{i,j}^T Q_{i,j} + \lambda I)^{-1} Q_{i,j}^T$  elde edilmesi (matris – matris çarpımı)
  - g.  $(Q_{i,j}^T Q_{i,j} + \lambda I)^{-1} Q_{i,j}^T \bar{r}(s_i, \bar{l}_j)$  vektörünün, diğer bir deyişle  $\bar{a}_{i,j}$  vektörünün elde edilmesi (matris – vektör çarpımı)
5. Sonuçta elde edilen kenar ağırlıkları matrisinin doskenara yazılması



Şekil 10 – Sistem Paralleştirme Adımları

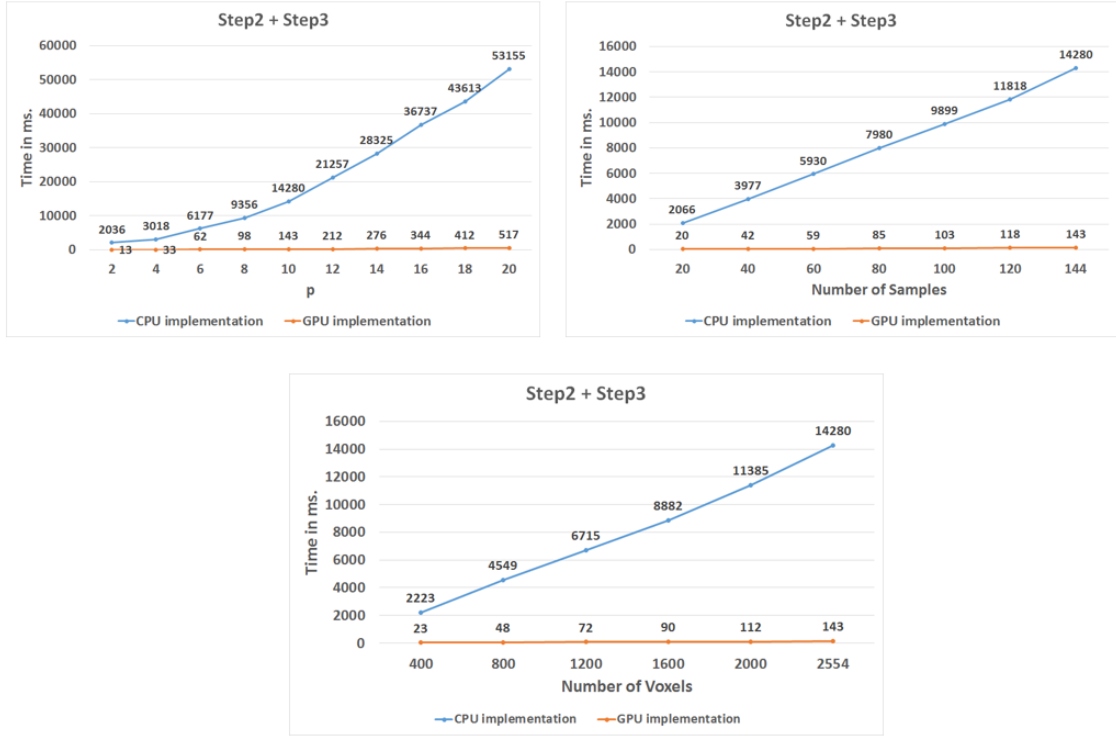
Performans analiz parametreleri 3 tane olup bunlar **örnek sayısı**, **voksel sayısı** ve **komşu sayısı (p)** dir. Diğer bir deyişle, bu değerler değiştiğçe sistemin performansının GPU ve CPU uygulamalarıyla nasıl değiştiği her bir adım için incelenmiştir. Aşağıdaki figürlerde her bir adım için GPU uygulamasının CPU uygulamasına göre hızlandırmasını yansıtan şekiller görülebilir:

**Adım 1:** Şekil 11, p-en yakın komşuların bulunması adımının komşu sayısına, örnek sayısına ve voksel sayısına göre CPU ve GPU kullanarak elde edilen sürelerin değişimini göstermektedir. Bu adım voksellerin xyz koordinatları kullanılarak elde edildiğinden, örnek sayısı arttıkça geçen sürede bir değişiklik gözlemlenmemektedir. Ancak, hem CPU, hem de GPU kullanıldığında voksel ve komşu sayısı arttıkça geçen süre artmaktadır. GPU kullanımı bu adım için harcanan süreyi CPU kullanımına göre oldukça azaltmıştır.



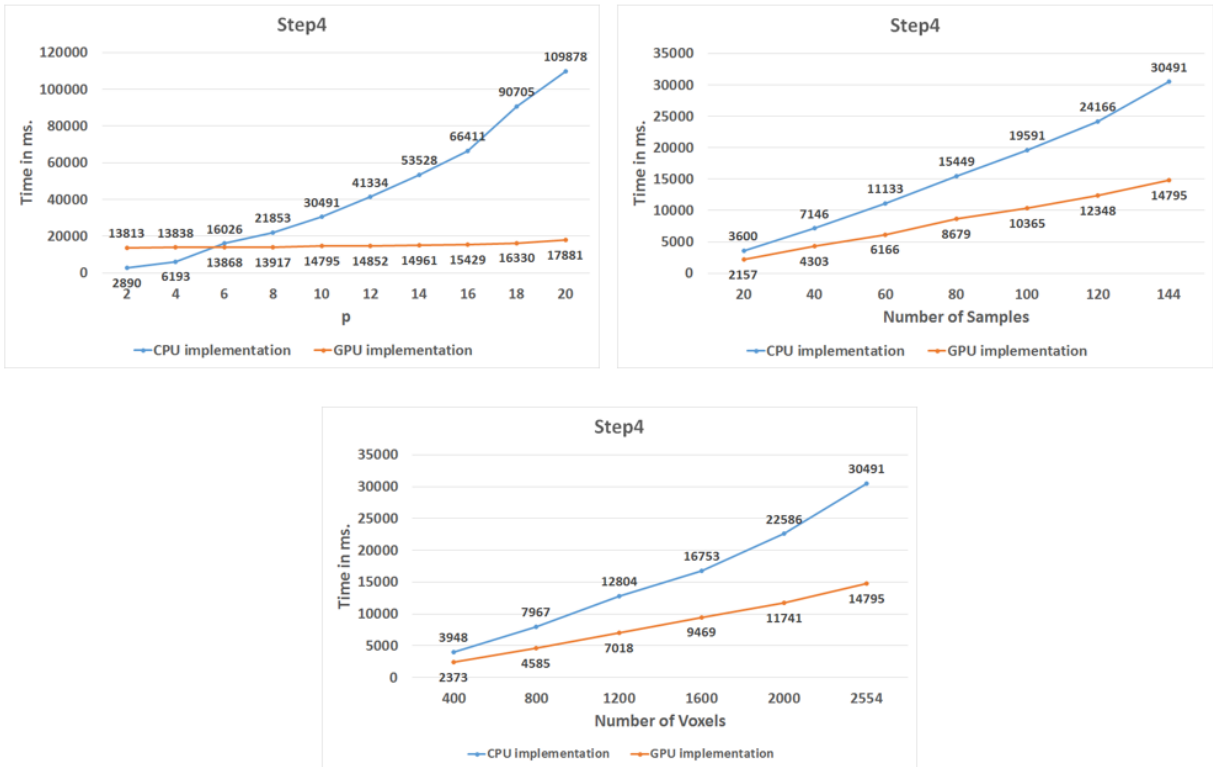
Şekil 11 – 1. Adım için GPU ve CPU kullanım sonuçları.

**Adım 2-3:** Geliştirilen uygulamada ikinci ve üçüncü adımlar bir arada hesaplandığı için bu adımların performansları beraber ölçülmüştür. Şekil 12'den görüleceği gibi 2 ve 3. Adımlar için komşu sayısı, voksel sayısı ve örnek sayısı arttıkça CPU ve GPU kullanarak elde edilen süreler artmaktadır. 1. Adımda olduğu gibi bu adımlarda da GPU kullanımı harcanan süreyi ciddi ölçüde azaltmıştır.



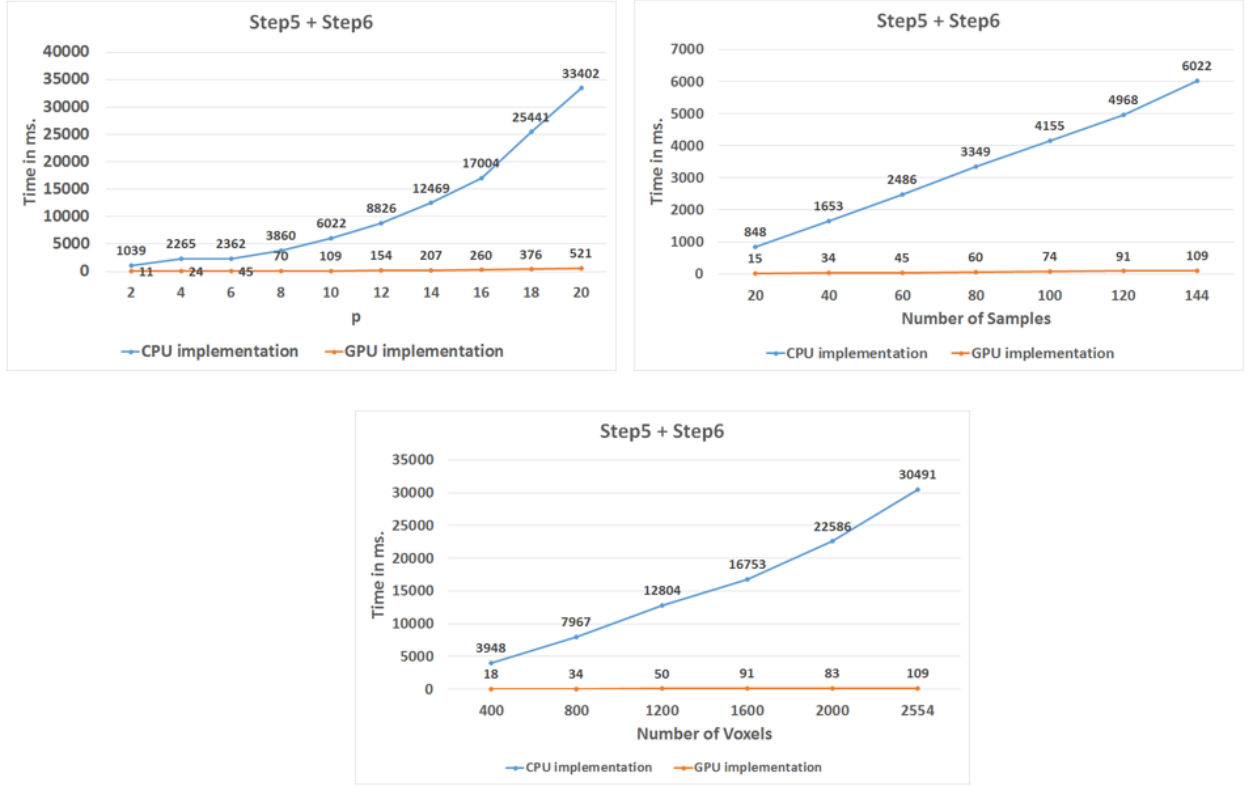
Şekil 12 – 2. Ve 3. Adımlar için GPU ve CPU kullanım sonuçları.

**Adım 4:** Bu adımda matris tersi alma işlemi CUDA CuBLAS kütüphanesi kullanılarak paralelleştirilmiştir. Komşu, voksel ve örnek sayısına göre CPU ve GPU süreleri Şekil 13’de görülmektedir.



Şekil 13 – 4. Adım için GPU ve CPU kullanım sonuçları.

**Adım 5-6:** Süreçte 5. ve 6. Adımlar beraber ele alınmıştır. Bu adımların sonunda final örgü kenar ağırlıkları elde edilmiştir. Komşu, voksel ve örnek sayısına göre CPU ve GPU süreleri Şekil 14’de görülmektedir.



Şekil 14 – 5. Ve 6. Adım için GPU ve CPU kullanım sonuçları.

Görüldüğü gibi, örgü kenar ağırlıklarının kestirilmesinin her bir adımı paralelleştirildiğinde harcanan süreler oldukça azalmıştır. Böylelikle, gelecekte daha çok voksel sayısı içeren, daha çok örnekli veriler için de bu paralel yapı kurulabilecektir.

### 3.8. CEREBRA: Proje Kapsamında Geliştirilen Görselleme Aracı ve Kullanıcı Arayüzü

İnsan beyni ve beynin işleyişini görselleyebilmek uzun zamandır araştırmacıların dikkatini çeken bir konudur. Ancak, günümüz nöro-görüntüleme sistemleri bir insanın hızlı ve verimli bir şekilde incelemesi için çok büyük veriler oluşturmaktadır. Bir insanın verimli çalışması için çalışmasının sonuçlarını hızlıca alabilmesi oldukça önemlidir. Bu sebeplerle, fMRG verilerinin görselleştirilmesi ve incelenmelerinin kolaylaştırılması ve hızlandırılması proje önerisinde de belirtildiği gibi projenin önemli adımlarından biridir.

Proje kapsamında geliştirilen CEREBRA yazılımı insan beyninin kompleks yapısını sadeleştiren ve bu yapıyı üç boyutlu görüntüye dönüştüren bir kullanıcı arayüzüdür. Bu amaç doğrultusunda fMRG cihazından elde edilen veriler önce YVA yöntemi ile modellenmektedir. Bunun için öncelikle üç boyutlu ortamda voksel yoğunluk değerleri görüntülenir, ardından



vokseller arasındaki bağlantıyı tahmin etmek için YVA modeli kullanılır. Son olarak da, voksellerden oluşan düğümler ve vokseller arasındaki ilişkileri gösteren kenar ağırlıkları 3 boyutlu bir beyin ağıнын görüntüsü olarak kullanıcıya sunulur. Beynin elde edilen üç boyutlu görüntüsü sayesinde, kullanıcı hangi durumlarda beynin hangi bölgelerinin ne kadar aktif olduğunu gözlemleyebilmektedir. Bu görüntü veriye bağlı olarak statik olabilir, tek bir anda çekilmiş FMRG verisini görselleştirebilir, veya voksel yoğunluk değerlerinin zaman içinde değişimini gösteren bir animasyon olabilir. Beyin görüntüsünün anlaşılabilirliğini arttırmak ve incelenmesini kolaylaştırmak için dinamik filtreleme yöntemleri de yazılıma eklenmiştir.

Yazılım geliştirme aşamaları ve detayları aşağıda belirtilmiş olan görevler şeklinde özetlenebilir

### **3.8.1. CEREBRA Yazılımı İçin Kullanılan Platform**

CEREBRA'nın kullanıcı arayüzü öncelikle Unity3D Game Engine kullanılarak oluşturulmuş fakat daha sonra QT Platformunda baştan tasarlanmıştır. Unity3D Game Engine ile hazırlanan yazılımda kullanılan C# kodları QT'de çalıştırılmak üzere C++ programlama dili kullanılarak tekrar yazılmıştır. Programın Unity3D Game Engine'den, QT'ye taşınmasındaki sebepler aşağıdaki maddelerde özetlenebilir:

- Grafik arayüzünün QT kullanılarak daha kolay hazırlanabilmesi: Unity3D Game Engine kullanıcı arayüzü hazırlanması ve kodların düzenlenebilmesi açısından pratik olsa da arayüz kod yazılarak hazırlanmakta ve görsel olarak iyileştirilmesi çok kolay olmamaktadır. Kullanıcı arayüzüne yeni bir tuş eklemek için mevcut koda ekleme yapılması gerekmekte ve yapacağı fonksiyon çağrılarının burada eklenmesi gerekmektedir. Bu durum arayüzün daha sonra düzenlenmesini zorlaştırmaktadır. Ayrıca, arayüzde yapılacak değişikliği görebilmek için bütün projenin çıktısının alınması gerekmekte olup dinamik bir şekilde bu değişikliği görmek mümkün olmamaktadır. QT kullanarak bu sorunlar ortadan kalkmıştır. QT kullanarak, arayüz programdan bağımsız olarak düzenlenebilir ve bu değişiklikler kod yazarak değil QT'nin sağladığı arayüz geliştirme aracı ile dinamik olarak yapılabilmektedir. Arayüz hazırlama ek Şekil 1'de görülebilir.
- Beyindeki çizgelerin görsellenmesindeki performans sorununun düzeltilmesi: Unity3D Game Engine'in aynı anda 3 boyutlu ve 2 boyutlu nesnelere çizememesinden dolayı 2 boyutlu olan çizgeler 3 boyutlu dikdörtgen prizmalar ile ekrana çizilmiştir. Bu yöntem ekrana çizilecek üçgen sayısını arttırdığı için yazılımın performansında ciddi düşüşler gözlemlenmiştir. Her ne kadar bu sorun dikdörtgen prizma yerine üçgen prizma kullanılarak giderilmeye çalışılsa da Unity3D Game Engine kullanılarak maksimum performans ve istenilen görüntü elde edilememiştir. QT kullanarak düğümler 3 boyutlu

vokseller olarak çizgiler de 2 boyutlu çizgiler olarak ekrana çizilebilmiştir. 2 boyutlu görüntüler beynin gösterildiği açığa göre dinamik olarak güncellenmektedir.

- Genel performans artışına olanak sağlanması: QT ile birlikte geliştirilen yazılımın programlama dili C++ olarak güncellenmiştir. Bu güncelleme C++ için geliştirilmiş olan OpenGL (Open Graphics Library) kütüphanesinin ve ekran kartı üzerinde çalışmayı sağlayan GLSL (OpenGL Shading Language) dilinin de kullanılmasına olanak sağlamıştır. Bu sayede beyin verisi üzerinde yapılacak değişiklikler veya ekran görüntüsü üzerinde gerçekleşecek olan değişimler ekran kartı işlemcisi tarafından daha kolay ve hızlı bir şekilde yapılabilmektedir. Önceden statik olarak yapılan bazı işlemler (voksel ve çizgelerin yoğunluklarına eşik değeri atama, vokselleri veya çizgeleri görüntüye ekleme veya kaldırma, voksel gruplarının belli renklere boyanması) artık interaktif bir şekilde kullanıcıyı bekletmeden yapılabilmektedir.

### **3.8.2. CEREBRA Grafik İşlemci Ünitesi ile Performans Arttırma**

Günümüzde hızla gelişen, özellikle zaman ve işlemci gücünden kazanmak için kullanılan paralel programlama teknikleri GPU hızlandırmalı hesaplama kütüphaneleri sayesinde standart bir bilgisayarda da uygulanabilir hale getirilmiştir. GPU hızlandırmalı hesaplama kütüphaneleri kullanılarak büyük veriler üzerinde uygulanacak işlemler grafik işlemci ünitesine aktarılarak işlemciye oranla çok daha hızlı yapılabilir. Bu noktada performans artışı öncelikle kullanılan verinin büyüklüğüne bağlı olup, grafik işlemci ünitesi (ekran kartı) modeline ve karşılaştırılan işlemciye göre değişiklik göstermektedir.

Örnek vermek gerekirse, beyin verisinde bulunan 40000 vokselin yoğunluk değerlerinin kontrol edilip belli bir değerin altındaki vokselleri görüntüden çıkartmak için geliştirilen bir algoritma GPU hızlandırılmalı hesaplama kütüphaneleri kullanılarak yazılır ve program ortalama bir dizüstü bilgisayarda çalıştırılırsa her bir vokselin kontrolü grafik işlemci ünitesinde o voksele ayrılmış bir çekirdek tarafından yapılır ve 40000 voksel eş zamanlı olarak işlenebilir. Aynı method, işlemci üzerinde çalışacak şekilde yazılır ve yine ortalama bir dizüstü bilgisayarda çalıştırılırsa 40000 voksel işlemcideki çekirdekler tarafından bölüştürülecek ve her çekirdek kendine düşen vokselleri sırayla işleyecektir. İşlemcinin aynı zamanda bilgisayardaki diğer programları ve işletim sistemini de ayakta tutması da hesaba katıldığı zaman aynı anda işlenen voksel sayısı işlemcideki çekirdek sayısı kadar bile olamayacaktır. Bunun yanında tek performans ölçütü çekirdek sayısı olmayıp verilerin aktarılma hızı, işlemcinin saat hızı, grafik işlemcinin işlem hızı gibi faktörler de performansı ciddi ölçüde etkilemektedir.

Geliştirilen araçta ekrana çizdirme işlemleri OpenGL (Open Graphics Library) kütüphanesi kullanılarak baştan yazılmıştır. Bu kütüphanenin kullanılması GLSL (OpenGL Shading Language) programlama dilinin de yazılıma dahil edilmesine olanak sağlamıştır. Bu programlama dili hem beyin görüntüsünün çizilmesinde çağırılan methodların optimize edilmesinde hem de beyin verisini işleyen methodların yazımında kullanılmıştır. Şu an programda hazır bulunan ve aşağıda bulunan görev paketinde anlatılan dinamik filtreler bu programlama dili kullanılarak geliştirilmiştir.

### 3.8.3. Dinamik Filtreleme Yöntemlerinin Geliştirilmesi

Kullanılan kütüphaneler ve yöntemler büyük beyin verisinin çok kısa zamanda işlenmesine olanak sağlamış, bu da daha önce statik olarak uygulanan filtrelerin dinamik (interaktif) olarak uygulanabilmesine yol açmıştır. Yani; bu filtreler sayesinde kullanıcı yaptığı ayarlamaların sonucunu ayarlamayı yaptığı sırada görebilmektedir. Hazırlanan dinamik filtreler ve uygulamaları aşağıdaki maddelerde detaylandırılmıştır.

- Voksel yoğunluk değerlerini normalleştirme: Voksel yoğunluk değerlerinin belli bir alt ve üst değerleri olmadığı ve bu değerler deneye, çekim yapılan cihaza, ve deneye giren bireye bağlı olarak değişebileceği için beyin verisi ilk basıldığı anda voksel aktiviteleri hakkında pek bilgi veremeyebilir. Bütün görüntü tek bir renk etrafında toplanmış halde gelebilir. Bu sorunu çözmek için kullanıcının atayacağı alt ve üst yoğunluk değerlerine göre gelen verideki yoğunluk değerleri normalize edilmektedir. Önceki sistemde bu değerler program tarafından otomatik bir şekilde atılırken yeni sürümde kullanıcının atayacağı değerlere göre yapılmaktadır. Bunun nedeni ise kullanıcının bu değerleri incelenen bireyin yoğunluk değerlerine veya test sırasında kullanılan makinenin yakalayabileceği en yüksek ve en düşük yoğunluk değerlerine göre atayabilme özgürlüğü sağlamaktır. Normalize edilmemiş bir görüntünün normalize edildikten sonraki değişimi ek Şekil 2 ve Şekil 3'te gösterilmektedir.
- Voksel yoğunluk değerlerine alt eşik değeri atama: Voksel yoğunluk değerleri normalize edildikten sonra görüntü iyileşse de voksel sayısındaki fazlalıktan dolayı bu görüntüden anlamlı bir bilgi çıkartmak veya aktif bölgeleri incelemek zor olabilir. Bu nedenle belli bir aktiviteye ulaşmamış vokseller bu method sayesinde elenebilir ve daha aktif çalışan bölgelerin çalışması daha kolay incelenebilir. Ek Şekil 4-10'da oksipital bölgedeki aktiviteyi daha kolay incelemek için çekilmiş örnekler bulunmaktadır.
- Voksel yoğunluk değerlerine alt ve üst eşik değeri atama: Bu özellik sayesinde belli bir yoğunluk aralığındaki vokselleri gizleyerek beyinde en çok ve en az çalışan vokselleri aynı anda görsellemek mümkün olmaktadır. Ek Şekil 5'de bu özelliğin çıktısı incelenebilir.

- Çizge yoğunluk değerlerini normalleştirme: Vokseller gibi çizge değerleri de deneye, deneyin yapıldığı makineye ve deneye giren kişiye göre değişiklik gösterebilir. Bu nedenle voksel yoğunluk değerlerini normalleştirme özelliği aynı işlevi yerine getirecek şekilde çizgeler için de eklenmiştir. Ek Şekil 6 ve Şekil 7’de normalleştirilmemiş çizgeler ve normalleştirilmiş çizgeler arasındaki görüntü farkı görülmektedir.
- Çizge yoğunluk değerlerine alt eşik değeri atama: Çizge sayısı voksel sayısının karesi ile orantılı olduğundan dolayı bütün çizgeleri bir anda görüntülemek ekrandaki görüntüyü anlaşılabilir bir hale getirmektedir. Bu filtre sayesinde daha zayıf bağlantıları gösteren çizgeler görüntüden kaldırılarak daha güçlü bağlantıların incelenmesi kolay hale getirilmektedir. Ek Şekil 8 ve Şekil 9’da eşik değeri atandıktan sonra elde edilen sadeleşmiş görüntü örnekleri mevcuttur.
- Çizge yoğunluk değerlerine alt ve üst eşik değeri atama: Alt eşik değeri atayarak en güçlü bağlantılar gözlenebilir ve iki voksel arasındaki benzerlikler ortaya çıkartılabilir. Bazı durumlarda, zayıf bağlantılar da nörobilimcilere önemli bilgiler sunabilir ve bu bilgiler bazı hastalıkların tanısında kullanılabilir. Bu nedenle geliştirilen yazılım güçlü bağlantılar ile zayıf bağlantıları da bir arada göstermektedir. Ek Şekil 10’da yeşil çizgiler zayıf bağlantıları gösterirken kırmızı çizgiler kuvvetli bağlantıları göstermektedir.

Hazırlanan yazılım 64-bit Windows 8 ve Windows 8.1 işletim sistemlerinde test edilerek çıktısı alınmıştır. Geliştirilen aracın çalışabilmesi için başka bir yazılıma ihtiyacı yoktur. Aşağıda verilen linkten indirildikten sonra test edilen koşulları sağlayan bir bilgisayarda “CEREBRA.exe” dosyasına tıklanarak çalıştırılabilir. Programın yanında nasıl çalıştığını anlamak ve veri yapısını anlamak için örnek veriler de verilmiştir. Ayrıca yazılım için hazırlanan kullanıcı kitapçığında da yazılımda bulunan bütün özellikler anlatılıp çeşitli örneklerle pekiştirilmiştir.

### 3.9. Hazırlanan Yazılımların Web Tabanlı Bir Paket Haline Getirilmesi

Bu projede geliştirilen yazılımlar [neuro.ceng.metu.edu.tr](http://neuro.ceng.metu.edu.tr) sitesi üzerinden tüm araştırmacılarla paylaşılmıştır. Site üzerinde “Codes” sekmesinden ulaşılabilecek olan programlardan FCL ön adı ile paylaşılan paketler 3.5 bölümünde anlatılan işlemleri ayrı programlar halinde yapmaktadır. MeshModel adı ile paylaşılan paket ise araştırmacıların kendi datalarını kullanarak **Yerel Voksel Ağları** modelini oluşturabilecekleri bir programdır. Bu paket yardımıyla, proje dahilinde oluşturulan ve detayları 3.5 bölümünde anlatılan bütün modeller MATLAB programı olarak araştırmacıların hizmetine sunulmuştur. CEREBRA adını verdiğimiz görselleme aracı yine aynı site üzerinden erişilebilir durumdadır.

#### 4. Sonuç

Proje dahilinde tasarımı ve katılımcılar üzerinde denenmesi tarafımızdan gerçekleştirilmiş olan verisetleri üzerinde, önermiş olduğumuz modellerin başarısı test edilmiştir. Elde edilen sonuçlar, voksellerin fonksiyonel komşuları ile yapmış olduğu doğrusal ilişkilerin öznitelik olarak kullanılmasının fMRG çıktısı olan ham vksel yoğunluk değerlerine göre bilişsel süreç sınıflandırmada daha başarılı neticeler verdiğini göstermektedir. İki farklı veriseti üzerinde uygulamış olduğumuz testler neticesinde, öne sürülen **Yerel Vksel Ağları** modelinin literatürde kullanılan diğer modellere göre istatistiksel olarak daha başarılı olduğu anlaşılmaktadır. Elde edilen başarılı sonuçlar neticesinde GPU programlama teknikleri ile model çıkarımının hızlandırılması, oluşturulan modelin insan-bilgisayar etkileşim sistemleri için de bir altyapı oluşturmasına olanak sağlamıştır. Geliştirilmiş olan model şu aşamada, yapılan deneyler üzerinde gerçek-zamanlı bir sonuç üretmese de dağıtık programlama ve büyük yapı yaklaşımları ile gerçek-zamanlı ve yüksek etiketleme performansına sahip bir sistem potansiyeli içermektedir. Proje dahilinde geliştirmiş olduğumuz görselleme aracı, sadece **Yerel Vokse Ağları** sonucu ortaya çıkan öznitelikleri değil, iki boyutlu bir vksel-değer matrisi üzerinde gösterilmiş tüm öznitelik değerlerini gösterebilecek şekilde geliştirilmiştir. İsteyen araştırmacılar, kendi özniteliklerini program üzerinde görselleyebileceği gibi, açık kaynak kod üzerinden eklemek istedikleri özellikleri kendileri gerçekleştirebileceklerdir.

## 5. Kaynaklar

- [1] Norman, Kenneth A., vd. "Beyond mind-reading: multi-voxel pattern analysis of fMRI data." *Trends in cognitive sciences* 10.9 (2006): 424-430.
- [2] Polyn, Sean M., vd. "Category-specific cortical activity precedes retrieval during memory search." *Science* 310.5756 (2005): 1963-1966.
- [3] Mitchell, Tom M., vd. "Learning to decode cognitive states from brain images." *Machine Learning* 57.1-2 (2004): 145-175.
- [4] Haynes, John-Dylan, and Geraint Rees. "Decoding mental states from brain activity in humans." *Nature Reviews Neuroscience* 7.7 (2006): 523-534.
- [5] Mitchell, Tom M., vd. "Learning to decode cognitive states from brain images." *Machine Learning* 57.1-2 (2004): 145-175.
- [6] De Martino, Federico, vd. "Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns." *Neuroimage* 43.1 (2008): 44-58.
- [7] Jin, Bo, vd. "Feature selection for fMRI-based deception detection." *BMC bioinformatics* 10.Suppl 9 (2009): S15.
- [8] Sun, Fengrong, vd. "Feature-space-based fMRI analysis using the optimal linear transformation." *Information Technology in Biomedicine, IEEE Transactions on* 14.5 (2010): 1279-1290.
- [9] Michel, Vincent, vd. "A supervised clustering approach for fMRI-based inference of brain states." *Pattern Recognition* 45.6 (2012): 2041-2049.
- [10] Wang, Defeng, vd. "Ellipsoidal support vector clustering for functional MRI analysis." *Pattern Recognition* 40.10 (2007): 2685-2695.
- [11] Chu, Carlton, vd. "Kernel regression for fMRI pattern prediction." *NeuroImage* 56.2 (2011): 662-673.
- [12] Cabral, Carlos, Margarida Silveira, and Patricia Figueiredo. "Decoding visual brain states from fMRI using an ensemble of classifiers." *Pattern Recognition* 45.6 (2012): 2064-2074.
- [13] Plumpton, Catrin O., vd. "Naive random subspace ensemble with linear classifiers for real-time classification of fMRI data." *Pattern Recognition* 45.6 (2012): 2101-2108.
- [14] Kuncheva, Ludmila I., vd. "Multi-modal biometric emotion recognition using classifier ensembles." *Modern approaches in applied intelligence*. Springer Berlin Heidelberg, 2011. 317-326.
- [15] M. Ozay, I. Oztekin, U. Oztekin, F. T. Yarman Vural, "Modeling cognitive states using machine learning techniques", *Neuroinformatics 2011*, Boston, MA, USA

- [16] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "Functional Mesh Learning for Pattern Analysis of Cognitive Processes", 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC), 2013.
- [17] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "Enhancing Local Linear Models Using Functional Connectivity for Brain State Decoding", International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 2014.
- [18] I. Onal, E. Aksan, B. Velioglu, O. Firat, M. Ozay, I. Oztekin, F. T. Yarman Vural, "Estimating Brain Connectivity for Pattern Analysis", 22nd IEEE Conference on Signal Processing and Communications Applications (SIU), 2014
- [19] I. Onal, E. Aksan, B. Velioglu, O. Firat, M. Ozay, I. Oztekin, F. T. Yarman Vural, "Modeling the Brain Connectivity for Pattern Analysis", 22nd International Conference on Pattern Recognition (ICPR), 2014.
- [20] B. Velioglu, E. Aksan, I. Onal, O. Firat, M. Ozay, F. T. Yarman Vural, "Functional Networks of Anatomic Brain Regions", 13th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC), 2014
- [21] Onal, Itir; Ozay, Mete; Vural, Fatos T.Yarman, "Modeling Voxel Connectivity for Brain Decoding," in Pattern Recognition in NeuroImaging (PRNI), 2015 International Workshop on , vol., no., pp.5-8, 10-12 June 2015
- [22] Bishop, Christopher M. Pattern recognition and machine learning. Springer, 2006.
- [23] Rubinov, Mikail, and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations." Neuroimage 52.3 (2010): 1059-1069.
- [24] Smith, Stephen M., vd. "Network modelling methods for FMRI." Neuroimage 54.2 (2011): 875-891.
- [25] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "A Mesh Learning Approach for Brain Data Modeling", IEEE 20th Conference on Signal Processing and Communications Applications (SIU), 2012.

## 6. Proje Dahilinde Yapılan Yayınlar

- [1] M. Ozay, I. Oztekin, U. Oztekin, F. T. Yarman Vural, "Mesh Learning for Classifying Cognitive Processes", arXiv:1205.2382v2.
- [2] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "A Mesh Learning Approach for Brain Data Modeling", IEEE 20th Conference on Signal Processing and Communications Applications (SIU), 2012
- [3] I. Onal, M. Ozay, O. Firat, I. Oztekin, F. T. Yarman Vural, "Information Distribution Analysis in the fMRI measurements with Degree of Locality Estimation", IEEE 21th Conference on Signal Processing and Communications Applications (SIU), 2013.
- [4] I. Onal, M. Ozay, O. Firat, I. Oztekin, F. T. Yarman Vural, "Analyzing the Information Distribution in the fMRI measurements by estimating the degree of locality", 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2013.
- [5] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "Functional Mesh Learning for Pattern Analysis of Cognitive Processes", 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC), 2013.
- [6] O. Ekmekci, O. Firat, M. Ozay, I. Oztekin, F. T. Yarman Vural, U. Oztekin, "Mesh Learning for Object Classification using fMRI Measurements", 12th IEEE International Conference on Image Processing (ICIP), 2013.
- [7] S. Alkan, F. T. Yarman Vural, "Localization of semantic category classification in fMRI images", 22nd IEEE Conference on Signal Processing and Communications Applications (SIU), 2014.
- [8] I. Onal, E. Aksan, B. Velioglu, O. Firat, M. Ozay, I. Oztekin, F. T. Yarman Vural, "Estimating Brain Connectivity for Pattern Analysis", 22nd IEEE Conference on Signal Processing and Communications Applications (SIU), 2014
- [9] O. Firat, M. Ozay, I. Onal, I. Oztekin, F. T. Yarman Vural, "Enhancing Local Linear Models Using Functional Connectivity for Brain State Decoding", International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 2014.
- [10] B. Velioglu, E. Aksan, I. Onal, O. Firat, M. Ozay, F. T. Yarman Vural, "Functional Networks of Anatomic Brain Regions", 13th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC), 2014.
- [11] I. Onal, E. Aksan, B. Velioglu, O. Firat, M. Ozay, I. Oztekin, F. T. Yarman Vural, "Modeling the Brain Connectivity for Pattern Analysis", 22nd International Conference on Pattern Recognition (ICPR), 2014.
- [12] Onal, Itir; Ozay, Mete; Vural, Fatos T.Yarman, "Modeling Voxel Connectivity for Brain Decoding," in Pattern Recognition in NeuroImaging (PRNI), 2015 International Workshop on , vol., no., pp.5-8, 10-12 June 2015



**TÜBİTAK**  
**PROJE ÖZET BİLGİ FORMU**

Proje Yürütücüsü:	Prof. Dr. FATOŞ TUNAY VURAL
Proje No:	112E315
Proje Başlığı:	Yerel Voxel Ağları İle Beyin Verisi Üzerinden Bilişsel Süreçlerin Modellenmesi Ve Otomatik Olarak Tanınması
Proje Türü:	1001 - Araştırma
Proje Süresi:	30
Araştırmacılar:	İLKE ÖZTEKİN
Danışmanlar:	
Projenin Yürütüldüğü Kuruluş ve Adresi:	ORTA DOĞU TEKNİK Ü. MÜHENDİSLİK F. BİLGİSAYAR MÜHENDİSLİĞİ B.
Projenin Başlangıç ve Bitiş Tarihleri:	01/04/2013 - 01/10/2015
Onaylanan Bütçe:	291485.0
Harcanan Bütçe:	188120.75

TÜBİTAK

<p>Öz:</p>	<p>Yapay zeka konusunda çalışan araştırmacılar, insan zekasından esinlenerek, insan zekasına benzeyen yapay sistemleri geliştirmeye çalışmaktadır. Temel amaç insan gibi düşünen, öğrenen ve problem çözebilen makineler yapabilmektir. Bu sistemlerle ilgili yazılım ve donanımları geliştirebilmek üzere birçok matematiksel yöntem ve algoritmalar geliştirilmiştir. Bu yöntemler sayesinde nesnelere algılama, tanıma, sınıflama ve öğrenme gibi önemli bilişsel süreçlerin matematiksel modelleri oluşturulmuştur.</p> <p>Bu projede, yapay zeka algoritmaları için geliştirilen yöntemleri kullanarak, bilişsel süreçleri modelledik. Diğer bir deyişle, bugüne kadar insan zekasını taklit ederek geliştirdiğimiz yapay zeka yöntemlerini tersinir bir mühendislikle geri çevirerek doğal zekayı modellemek üzere kullandık. Modellerimizi bilişsel süreçler sırasında kaydedilen fMRG görüntüleri kullanarak oluşturduk. Bunun için, bir dizi bilişsel deney tasarladık ve bu deneyleri denekler üzerinde uygularken fMRG sinyallerini kaydettik. Böylece, elde ettiğimiz etiketli öğrenme verilerini geliştirdiğimiz makine öğrenme algoritmalarını eğitmek için kullandık.</p> <p>Projenin en önemli çıktılarından birisi de oluşturulan modellerin ve elde edilen fMRG verilerinin, web tabanlı bir ortamda tüm araştırmacıların hizmetine açılmasıdır. Böylece oluşturduğumuz metodoloji ve programları nörobilimciler veri analizinde kullanabileceklerdir. Geliştirdiğimiz matematiksel beyin modeline Yerel Voksel Ağları (YVA) adını verdik. İnsan-bilgisayar etkileşimi teknolojileri için bilimsel bir altyapı oluşturma potansiyeline sahip olan bu model, iki farklı veri seti üzerinde gerçekleştirilen deneylerde, literatürde yoğun olarak kullanılan diğer Çoklu Voksel Örüntü Analizi (ÇVÖA) yöntemlerine göre daha başarılı sonuçlar ortaya koyduğu anlaşılmıştır.</p> <p>Yerel Voksel Ağları yöntemi, fMRG sinyallerinin en küçük birimi olan voksellerden elde edilen zaman serileri arasındaki ilişkiyi doğrusal denklemlerle modellemektedir. İnsan beyinde, birbirine yakın nöronların benzer aktiviteler gösterdiği bilinmektedir. Bu yerel benzerlik, birbirine yakın vokseller arasındaki ilişkinin doğrusal olduğunu göstermektedir. Bazen de birbirinden uzak vokseller de beyindeki direkt yollar aracılığı ile benzer aktivasyonlara sahip olabilmektedir. Bu tür vokseller yerel olarak değil de fonksiyonel olarak birbirine komşu kabul edilebilir. Geliştirmiş olduğumuz bu yeni komşuluk sistemine fonksiyonel komşuluk adını verdik. Ve vokseller arasındaki doğrusal ilişkiyi, Fonksiyonel Komşuluk adını verdiğimiz bu ölçütü kullanarak modelledik. YVA modelinin, fMRG sinyallerinden gelen ham özniteliklere göre, bilişsel süreçleri etiketleme konusunda daha başarılı olduğunu deneysel olarak gösterdik.</p> <p>Proje kapsamında öncelikle fMRG verisi toplamak üzere bir dizi deney tasarladık ve bu deneyleri Bilkent Üniversitesi UMRAM merkezinde gerçekleştirdik. Bunun için insan beyinin bilgileri nasıl depolandığını ve nesnelere nasıl tanıdığını anlamak üzere 3 değişik deney yaptık. Bu deneylerde değişik nesnelere tanıdığı etiketli fMRG verileri elde ettik. Daha sonra verileri işleyerek gürültülerden arındırmaya çalıştık ve modelimiz için uygun hale getirdik. Kullandığımız görüntü iyileştirme teknikleri ile, YVA yönteminde elde edilen etiketleme performansını arttırdık.</p> <p>YVA yöntemi sonucunda elde edilen beyin ağını kullanarak birçok öznitelik oluşturulabilir. Biz bu çalışmada YVA'dan elde ettiğimiz kenar ağırlıklarını öznitelik olarak kullanarak Yapay Sinir Ağları, Destek Vektör Makinaları ve k-En Yakın komşuluk gibi çeşitli sınıflandırıcıları eğittik. Bunların içinde en başarılı olanları seçerek performansları ölçtük.</p> <p>Yöntemin başarılı sonuçlar vermesi neticesinde, doğrusal ilişki özniteliklerinin çıkarılma adımının hızlandırılması için GPU programlama tekniklerinden faydalandık. Son olarak, ortaya çıkarılan özniteliklerin beyin modeli üzerinde bilim insanlarına sunulması için, bir kullanıcı ara yüzü geliştirdik.</p>
<p>Anahtar Kelimeler:</p>	<p>Makine Öğrenmesi, Örüntü Tanıma, Yapay Zeka, Beyin Okuma, İnsan-bilgisayar etkileşimi, fMRG, GPGPU</p>
<p>Fikri Ürün Bildirim Formu Sunuldu Mu?:</p>	<p>Hayır</p>

Projeden Yapılan Yayınlar:	<ol style="list-style-type: none"><li>1- Enhancing Local Linear Models Using Functional Connectivity for Brain State Decoding (Makale - Diğer Hakemli Makale),</li><li>2- Modeling the Brain Connectivity for Pattern Analysis (Bildiri - Uluslararası Bildiri - Poster Sunum),</li><li>3- Estimating Brain Connectivity for Pattern Analysis (Bildiri - Ulusal Bildiri - Sözlü Sunum),</li><li>4- Localization of semantic category classification in fMRI images (Bildiri - Ulusal Bildiri - Sözlü Sunum),</li><li>5- FUNCTIONAL NETWORKS OF ANATOMIC BRAIN REGIONS (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</li><li>6- Large Scale Functional Connectivity For Brain Decoding (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</li><li>7- Functional Mesh Learning for pattern analysis of cognitive processes (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</li></ol>
----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

TÜBİTAK