Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 57-63, Ekim 2019
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 57-63, October 2019
Copyright © 2019 EJOSAT
Research Article

Yüz İfadesine Göre Satranç Oynayan Robot: Çolak

Şafak Kayıkçı*

Abant İzzet Baysal Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Bolu, Türkiye (ORCID: 0000-0002-3325-4731)

(Bu yayın HORA 2019 kongresinde sözlü olarak sunulmuştur.)

(İlk Geliş Tarihi 1 Ağustos 2019 ve Kabul Tarihi 22 Ekim 2019)

(**DOI:** 10.31590/ejosat.636329)

ATIF/REFERENCE: Kayıkçı, Ş. (2019). Yüz İfadesine Göre Satranç Oynayan Robot: Çolak. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 57-63.

Öz

Bu çalışma kapsamında, Arduino Braccio Robot kolun rakibinin yüz hatlarından ölçeceği duygu yoğunluklarına göre Stockfish satranç motoruna hamleler göndererek satranç oynaması amaçlanmıştır. Uygulamanın kapsamında ortam tanıma, rakibi izleyebilecek görüntü sistemine ve hareketlerin gerçekleştirilmesi için gerekli yeteneklerin kazandırılması gerekmektedir. Bu gerekli tanıma, hareket işlemlerinin yapılması için kişisel bilgisayarlar üzerinde çalıştırılacak bir yazılım hazırlanmıştır. Robot koluna verilecek komutlar, Arduino Yun üzerinden gönderilir. Kolun, ortamın ve rakipten alınacak durum bilgileri ise satranç tahtasına paralel ve rakibe paralel olarak yerleştirilecek ve Arduino ile uyumlu olarak çalışacak olan iki adet kamera ile sağlanacaktır. Çalışmanın amacı, tamamen kendi kendine gerekli algoritmalar ve yetenekler ile satranç oynayan bir robot kol inşa etmek. Bunun için gerekli kinetik ve görüntü işleme çalışmalarının yapılmasıyla böyle bir sistemi meydana getirilmesinin mümkün olduğunun gösterilmesidir.

Anahtar Kelimeler: Robotik, Görüntü işleme, Satranç

Robot Playing Chess by Facial Expression: Colak

Abstract

In this study, it is aimed to play chess by sending moves to Stockfish chess engine according to emotion intensities of Arduino Braccio Robot arm's opponent's facial features. Within the scope of the application, it is necessary to gain the necessary skills for the recognition of the environment, the visual system that can monitor the competitor and the realization of the movements. In order to carry out the necessary recognition and movement procedures, a software has been prepared to be run on personal computers. The commands to be given to the robot arm are sent via Arduino Yun. The status information of the arm, the environment and the opponent will be placed parallel to the chessboard and parallel to the opponent and provided with two cameras which will work in harmony with Arduino. The aim of the study is to build a robotic arm that plays chess completely with the necessary algorithms and capabilities. It is shown that it is possible to create such a system by performing necessary kinetic and image processing studies.

Keywords: Robotics, Image processing, Chess

* Sorumlu Yazar: Bolu Abant İzzet Baysal Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye, ORCID: 0000-0002-3325-4731, safak.kayikci@ibu.edu.tr

1. Giriş

Satranç birçok insanı, hatta oynamayı sevmeyenleri büyülemektedir. Her yaşta, her ülkede, dine veya etnik kökene dikkat edilmeden oynanır. Akıl ve iyi karar vermenin bir sembolü olarak kullanılır. Satrançın doğası literatürde sıklıkla tartışılmaktadır. Sadece bir oyun değil, aynı zamanda bir spor (rekabet unsurundan ötürü), bir sanat (izin verdiği güzel kombinasyonlar nedeniyle) ve bir bilim (çalışılan sistematik bir yol nedeniyle) olarak sunulmaktadır. Satranç bir çok bilimsel araştırmanın konusu olmuştur. Sosyoloji, etnoloji, felsefe, matematik ve sinirbilimi içeren bir dizi akademik disiplinler tarafından araştırılmıştır. Bilgisayar bilimi de, başta yapay zeka olmak üzere satrancı derin biçimde incelemektedir. Yapay zekâda satranç, makine öğrenmesi ve arama algoritmalarının geliştirilmesi için standart bir unsur olmuştur. Psikolojide ise algı, hafıza, öğrenme, düşünme ve karar verme konularında araştırma konusudur.

Satranç, 8x8'lik bir ızgarada düzenlenmiş 64 kareden oluşan damalı bir tahtada oynanan iki oyunculu bir strateji oyunudur. [1] Oyun dünya çapında milyonlarca insan tarafından oynanır. Satranç'ın, 7. yüzyıldan bir süre önce Hint oyun chaturanga'sından türetildiğine inanılmaktadır. Robotik oyunlar, insan robot etkileşimi alanında popüler araştırma konuları olarak ortaya çıkmaktadır. Bu oyunlar arasında da satranç her zaman araştırmacıların ilgisini çekmiştir. Satranç oyunu, sürekli olarak tahtayı ve parçaları gözlemlemek, durumları düşünmek ve rakip ile eşgüdüm içerisinde olmak gibi çeşitli zorluklar içerir.

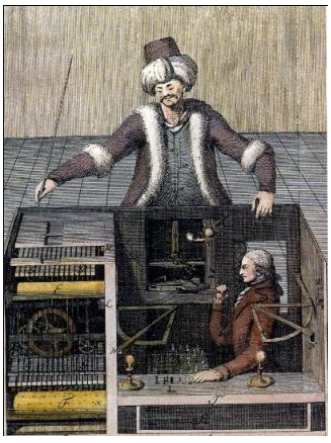
Bu çalışmada, bir eğitim robot kolu olan Arduino Braccio Robotic Arm [1] ünitesine rakibin duygu yoğunluklarına göre satranç hamleleri yaptırılması amaçlanmıştır. Gerçekleştirilecek hareketler , aynı düzlem üzerinde, farklı karelerde bulunan taşların bir kareden bir diğerine aktarılmasıyla oluşturulmuştur. Geliştirilen yazılım, kameradan aldığı görüntüleri kullanarak, boş tahtanın sınırlarını ve gerekli diğer özelliklerini kaydettikten sonra, sırasıyla taşları ve dolu kareleri tespit eder. Bu aşmada, taşın, karenin neresinde olduğu da hesaplanır. Rakibin yaptığı hamleden sonra , taşın yerden alınması ve nereye konulacağı sağlanmalıdır. Taşın taşınması sırasında, diğer taşlara çarpılmaması gerekmektedir. Bu gibi kısıtlamalar, robt kolun hareket algoritmalarında göz önünde alınmıştır.

2. Tarihçe ve Önceki Çalışmalar

Robotların insana benzeyen, ancak bazı yönleriyle insandan eksik olan varlıklar olduğu düşüncesi çok eski bir düşüncedir. İlk sibernetikçi kabul edilen Ebul-iz İsmail bin ar-Razzaz el-Cezeri 1205-1206 yıllarında yazdığı "Kitab-ül'-Camü Beyne'l-İlmi-i ve'l-amelen-Nafi' Fi Sınaati'lHiyel" adlı kitabın içinde, 300'e yakın otomatik makine ve sistemleri ile ilgili bilgi verdikten sonra çalıma özelliklerini şemalarla göstermiştir [1].

Robotikteki çoğu ilerleme ise hem kurgusal hem de gerçek hayat alanında 20. Yüzyılda gerçekleşmiştir. 1921 yılında Karel Capek isimli Çek yazar R.U.R. (Rossum's Universal Robots – Rossum'un Evrensel Robotları) isimli oyununda ilk kez "Robot" tanımını kullanmıştır. Robot kelimesi Çekçe orijinalinde "zorunlu iş" anlamına gelmektedir. R.U.R. resmi olarak Robot teriminin tarihte kullanıldığı ilk yayındır [2].

Endüstride ilk robot (UNIMAIE) 1961'de General Motors'un New Jersey'deki araba fabrikasında devreye girmiş, 60'lı yıllarda iki üniversite, Stanford ve MIT, robot araştırmalarında başı çekmiş, daha sonra birçok özel kurulu ve üniversite robotlarla ilgili çalışmalara başlamıştır. Satranç oynayan bir robotun ilk örneği, 1769 yılında yapılmış ve 1770 yılında sergilenmiştir. Bu makine, gerçek bir robot değildi fakat izleyicilerin öyle olduğunu zannetmesi sağlanıyordu. "Satranç oynayan Türk otomat" olarak tanımlanan bu makine, 120 cm. uzunluğunda, 105 cm. genişliğinde ve 60 cm. yüksekliğinde akça ağaçtan ve üzerine satranç tahtası çizilmiş tekerlekli bir kabinet önünde oturan bıyıklı, türbanlı ve pelerinli bir Türk figüründen oluşuyordu. Öndeki kapak açılıp dolabın ve Türk'ün içine bakıldığında irili ufaklı pek çok kaldıraç, makara ve başka karmaşık mekanik sistemler görülebilmekteydi [3].



Şekil 1: Satranç oynayan Türk otomat

2006 yılında, Ortadoğu Teknik Üniversitesi Enformatik Enstitüsü'nde "Satranç Oynayan Robot Sistemi" isimli proje, Musa ATAŞ tarafından, Ferda Nur ALPARSLAN'ın danışmanlığında gerçekleştirilmiştir. Proje, TÜBİTAK MAG(Mühendislik Araştırma Grubu) tarafından desteklenmiştir. Sistem, görüntü sistemi, satranç yazılımı, robot kolu ve ana kontrol programından oluşmaktadır. Görüntü

sistemi, periyodik olarak almakta olduğu görüntüleri önce görsel gürültülerinden arındırmaktadır. Elde edilen bu temiz görüntüden satranç 9 tahtasının elde edilmesi için, satranç tahtasının dış sınırları, köşe tespit ve Sobel kenar tespit algoritmaları kullanılarak belirlenmektedir. Aynı görüntüden, satranç karelerinin sınırları da belirlenilmektedir. Satranç taşları, özel olarak tasarlanmışlardır ve "yapay sinir ağları" çalışması ile sisteme tanıtılmıştır. Görüntü alan kamera, sabit bir yerde durmaktadır. Hamle yapıldığının anlaşılması için, eldeki görüntü ile bir önceki geçerli görüntüdeki farklar hesaplanılmaktadır. Rakibin hamlesine karı hamle üretmek için, bir satranç programı kullanılmıştır. Üretilen hamlenin, robot kolu tarafından yerine getirilmesi ise, "ters kinematik" hesaplarla gerçekleştirilmektedir. Yazılım dili olarak C ve Java kullanılmıştır. 2002 yılında, Ecole Polytechnique'de(Montreal, Fransa), "Autonomous Chess-playing Robot" isimli çalıma, Timothée COUR, Rémy LAURANSON ve Matthieu VACHETTE tarafından gerçekleştirilmiştir. Bu çalışmada, insana karı satranç oynayan bir robot kol sistemi tasarlanmıştır. Kullanılan robot kolu, 4 serbestlik derecelidir. Robotun bilgisayarla balantısı, bir ISA kart ile sağlanmıştır. Temel giriş/çıkış fonksiyonları, Analog-Dijital çeviriciler ile sağlanmıştır. Robottaki servo motorlardan alınan geri besleme, kolun düzgün hareket ettirilmesi için kullanılmıştır. Alınan görüntü, 512x512 boyutunda ve 8 bitlik gri tonlamalıdır. Alınan bu görüntü üzerinden, karelerin kartezyen koordinatları hesaplanmakta, bir kare üzerinde ta olup olmadığı tespit edilmekte ve dolu bir karedeki taın rengi ve tipi elde edilmektedir. Görüntü analizleri için, "Fast-Fourier" Dönüşüm algoritmaları, Yapay sinir ağları, karakteristik bölge analizi, filtreleme ve şablon uyumu çalışmaları yapılmıştır. Sonuç olarak, karelerin tespiti için köşe tespit algoritmaları kullanılmış, karelerin içerikleri hakkında bilgi edinmek için de karenin "ortalama gri değeri"ni kullanan toplam ve fark 10 tanımlayıcıları belirlenmiştir. Bu tanımlayıcılar ile, ne renk bir kare üzerinde ne renk bir ta olduğunun tespiti yapılabilmiştir. Taşların çeşitlerinin belirlenmesi, yoğun işlem gerektirecek bir işlem olarak ortaya çıktığı için, başlangıcı bilinen bir dizilimin her bir hamle sonunda ne şekilde değişeceğinin, yalnızca son hamlenin hangi kareden hangi kareye yapıldığının bilinmesi ile hesap edilebileceği gerçeği vurgulanmıştır.

2006 yılında, Kahire Üniversitesi Bilgisayar Mühendisli Bölümü'nde, A. K. Farahat, A. M. Hassan ve M. A. El-Nagar, "A Vision System for Chess Playing Robots" isimli çalışmayı gerçekleştirmişlerdir. Bu çalışmada, bir bilgisayarın, insan karısında gerçek bir satranç takımı ile satranç oynayabilmesine olanak veren bir sistem geliştirilmiştir. Sistem, satranç tahtasını, bir kamera ile izlemektedir. Sistem, "Yapay Zeka" ile desteklenmiştir ve oyunu oynayabilmek için bir robot kolu kullanmaktadır. Sistemin görsel işlemi, kameradan aldığı resimde satranç tahtasının dört köşesini bulmakla başlar. Bunun için, Harris ve Stephens tarafından 1988 yılında önerilen "Plessey feature point detector" köşe bulma algoritması kullanılmıştır. Bulunan bu köşeler kullanılarak, tahta görüntüsü, görüntünün tamamından ayrılır ve perspektif düzeltme yapılarak elde edilir. Oyunun, bilinen bir dizilim ile balaması gereklidir. Yapılan hamleler, hamleden önceki ve hamleden sonraki görüntülerin karşılaştırılması ile tespit edilir. Alınan görüntülerden, hangi karenin dolu, hangi karenin boş olduğu bilgisi ve dolu karelerdeki taşların rengi de elde edilmektedir. Renk bilgisi, gri tondaki resimdeki her karenin histogram bilgisi incelenerek elde edilmektedir. Dolaysıyla elde edilen bilgi, taşın gerçek rengi değil, oyundaki "siyah" taşlardan mı yoksa "beyaz" taşlardan mı olduğunun bilgisidir. Sistemin satranç mantığı için, "GNU chess" programı kullanılmıştır. "GNU chess", açık kaynak kodlu bir programdır. Kullanılan robot kolu, 4 serbestlik derecelidir. Robot kolunun hareketleri hesaplanırken, ters kinematik hesaplamalar 11 kullanılmıştır. Çalışmanın sonunda, sistemin denenmesi için yapılan 90 adet hamlenin 89'u, sistem tarafından doru olarak tespit edilmiştir. Robot kolunun hareketleri de tatmin edici görülmüştür fakat robota uzak karelerde, taşın taşınması sırasında, diğer karelerdeki taşlara istenmeyen temaslar olduğu da görülmüştür.

2. Materyal ve Yöntem

Uygulanan yöntemler dört ana başlık altında toplanmıştır. İlki, kamera ile satranç tahtasının birbirine paralel olarak konumlandırılmasının ardından kameranın aldığı görüntüler ile satranç tahtası karelerinin köşe noktalarını tespit etmesi ve bu karelerin isimlendirilmesi (a1,b7,c3..vs) aşamasıdır. İkinci ana işlev ise taşların hangi kategoride olduğunun tanımlanması ve satranç tahtası üzerinde bulunan satranç taşlarının yerlerinin tespit edilmesidir. Üçüncü aşama robot kolunun satranç tahtası üzerindeki hareketleri ve taşları taşıması aşamasıdır. Dördüncü ve son bölüm, yüz tanıma sayesinde Stockfish contempt değerlerinin, duygulara bağlı olarak değişikliğini sağlamaktır.

2.1. Kullanılan Açık Kaynaklı Yazılımlar : OpenCV ve Stockfish

OpenCV, bilgisayarlı görü, resim işleme ve makine öğrenmesi için lider açık kaynaklı bir kütüphanedir ve şimdi gerçek zamanlı çalışma için GPU hızlandırma özelliğine sahiptir [4]. BSD lisansı altında yayınlanmıştır ve bu nedenle hem akademik hem de ticari kullanım için ücretsizdir. C ++, C, Python ve Java arayüzlerine sahiptir ve Windows, Linux, Mac OS, iOS ve Android'i destekler. OpenCV, hesaplama etkinliği için ve gerçek zamanlı uygulamalara güçlü bir odaklanma ile tasarlanmıştır. En iyi duruma getirilmiş C / C ++ ile yazılmış kütüphane çok çekirdekli işlemlerden faydalanabilir. Tüm dünyada kabul edilen OpenCV, 47 binden fazla kullanıcı topluluğuna ve 6 milyondan fazla tahmini indirme sayısına sahiptir. OpenCV'nin projesinin temel hedefleri, temel vizyon altyapısı için sadece açık değil, aynı zamanda optimize edilmiş kodlar sağlayarak gelişmiş vizyon araştırması; geliştiricilerin üzerine inşa edebileceği ortak bir altyapı sağlayarak vizyon bilgisini dağıtılarak kodun daha kolay okunabilir ve aktarılabilir olması; taşınabilir, performansı optimize edilmiş kodu ücretsiz olarak kullanıma sunarak, kodun açık veya serbest olmasını gerektirmeyen bir lisansla, ileriye dönük vizyon tabanlı ticari uygulamalar geliştirilmesi oluşturmaktadır.

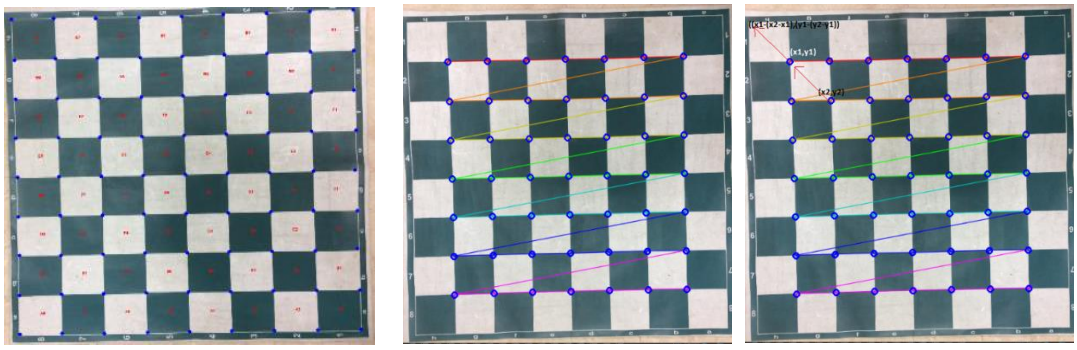
Stockish, çeşitli masaüstü ve mobil platformlarda kullanılabilen, ücretsiz ve açık kaynaklı bir satranç motorudur [5]. Marco Costalba, Joona Kiiski, Gary Linscott ve Tord Romstad tarafından geliştirilen ve açık kaynaklı geliştiriciler topluluğunun bir çok katkısı ile geliştirilmiştir. Stockish, çok işlemcili sistemlerde 512 CPU iş parçacığı kullanabilmektedir. Aktarım tablasının azami boyutu 128 GB'dir. Stockish gelişmiş bir alfa-beta araması gerçekleştirir ve bitboardlar kullanır. Diğer motorlarla karşılaştırıldığında, kısmen daha agresif budama ve geç hareket azaltma nedeniyle büyük arama derinliği ile karakterizedir. Stockfish C ++ dilinde yazıldığından, Android, iOS, Linux, OS X ve Windows gibi çeşitli işlemciler ve işletim sistemleri için derlenebilir ve oluşturulabilir. Macintosh için

Stockish, Stockish web sitesinden de sorumlu olan Daylen Yang tarafından yaptırılmıştır. İOS için Stockish, Tord Romstad tarafından yaptırılmıştır.

2.2. Tahtanın Tanıtılması ve Köşe Tespiti

Program, başlangıcında, tahtanın tamamını görüntülemeye ihtiyaç duyar bu sebepten dolayı kameranın, tahtanın tamamını görmeye ihtiyacı vardır ve tahtanın kareleri tamamen boş olmalıdır. Tahta boş iken kameradan alınan bu görüntü, analiz edilerek tahtanın kaç kareden oluştuğu belirlenir, boş karelerin koordinatları hesaplanır. Kamera aynı konumda iken taşlar yerlerine dizilir ve bu sefer de tahtaya taşlar konumlandırılmışken görüntü alınır. Bu görüntü ile ilk görüntü arasındaki farklılıklar kullanılarak dolu kareler ve dolu karelerdeki taşların konumları tespit edilir. Satranç tahtasına tam sayı ile dizilmiş olan rengi rakip temsil eder ve ilk hamleyi gerçekleştirmek üzere de robot kolumuz Stokfish'ten aldığı değeri oynayarak oyunu başlatır. Oyun başladığı andan itibaren rakibin yüzü bilgisayarımızın kamerası ile gerçek zamanlı olarak izlenir ve her 10 saniyede bir aldığı duygu bilgisi ile Stockfish contempt değeri belirlenerek robot kol oyunu gelen değerlere göre oynar.

Köşe tespiti yapmak için OpenCv'nin "findChessboardCorner" fonksiyonunu kullanılmıştır. Bu fonksiyon, giriş görüntüsünün satranç tahtası modelinin bir görünümü olup olmadığını ve iç satranç tahtasının köşelerini tespit edip etmediğini belirlemeye çalışır. Bu noktada satranç tahtası modelindeki pixel yoğunluğuna bakarak beyaz kareden siyah kareye geçişi veya siyah kareden beyaz kareye geçildiğini kontrol eder. Eğer böyle bir geçiş varsa köşe olarak algılar. Köşelerin tümü bulunursa ve belirli bir sıraya (satırdaki satır, her satırda soldan sağa) yerleştirilirse, fonksiyon sıfır olmayan bir değer döndürür. Aksi takdirde, işlev tüm köşeleri bulmakta başarısız olursa veya bunları yeniden sıralayamazsa, sıfır değerini döndürür. Bulunan noktalar işaretlenir ve işaretlenen noktalar yine OpenCv'nin "cornerSubPix" fonksiyonuyla yerleri hesaplanarak döndürülür.



Şekil 2: OpenCV ile köşe ve kare tespiti

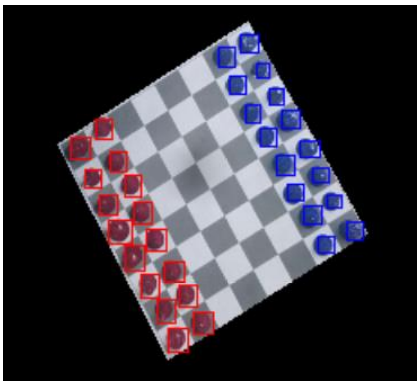
Fonksiyon ile iç noktalar tespit edildikten sonra dış köşeleri bulmak için ise "cornerSubPix" fonksiyonundan köşe noktalarını alıp bu noktalara göre dış köşelere öteleme yaparak noktaları tanıtılmıştır. Şekil 2'deki gibi her bir nokta üst köşelerindeki iki çapraz noktadan farkını çıkararak bulunmuştur. Bulunan noktaların ortasına OpenCv'nin "FONT_HERSHEY_SCRIPT_SIMPLEX" fonksiyonu ile yazılan kare isimlerinin fontunu belirlenmiştir. Ardından yine OpenCv'nin "putText" fonksiyonu ile köşelerin orta noktasına kare isimlerini yazılmıştır, orta noktaları bulmak için ise bir nokta için

$$((x2-x1)/2, /(y2-y1)/2)$$

şeklinde her bir köşe için orta nokta tanımlaması yapılmıştır.

2.3. Satranç Taşlarının Tespiti

Satranç taşları için kırmızı ve mavi renkler kullanılmıştur. Bunun sebebi, satranç taşlarının renk özelliğine göre tanımlanmasından dolayıdır. Kırmızı renkli taşlar , satrançtaki siyah taşları ifade etmekte iken mavi taşlar ise beyaz taşları ifade etmektedir. Bu tanıma işlemi, OpenCV'de RGB(Red,Green,Blue) ve HSV(Hue,Saturation,Value)'ye dönüştürülmüştür. RGB'yi HSV'ye döndürme nedeni ise bilgisayarın görüntü işleme uygulamasında belirli bir renkteki nesneyi ayırt etmek istenildiğinde HSV renk uzayını kullanılması daha elverişli olmasıdır. Çünkü HSV'deki Hue değeri ile kullanılan rengin eşik değerlerini ayarlanabilmektedir. Şekil 3'te, belirtilen eşik değerleri arasındaki renkleri(taşların) kare içine alarak her bir taş belirtilmiştir. OpenCV'nin "findContours" isimli fonksiyonuna belirtilen eşik değeri ve bu eşik değerine göre kare içine alınacak renkler için gerekli parametreler gönderilmiştir. Kare içine alınan taşların orta noktalarını alınıp ve daha öncesinde satranç tahtasının her bir karesinin köşelerinin koordinatları ile karşılaştırma yapılmıştır. Sonucunda ise taşlarının orta noktalarının hangi köşe koordinatları arasında olduğunu bulunarak ,taşların hangi satranç karesi(a1,c3,d2… vb) içinde olduğunu tespit edilmiş ve bunlar bir liste halinde tutulmuştur.



Şekil 3: Taşların renklere göre tanınması

Gerçek zamanlı olarak satranç tahtası üzerinde hareket algılamasını kontrol edilmiştir. Bunun sebebi, taşların üzerinde bir hareket veya herhangi bir taşı görülmediği zaman "control" adında oluşturulan fonksiyon ile 10 saniye boyunca rakibin hamle yapmasını beklenmesidir. 10 saniye sonunda fonksiyonda satranç taşlarının eski yerlerinin listesi ile yeni yerlerinin listesi karşılaştırılmakta ve farklılık olan dizi indekslerini Stockfish'e gönderilmektedir. Eğer iki farklı değişiklikte direk bulunduğu konumları Stockfish'e göndermektedir fakat Rook hamlesinde ise 4 farklı durum vardır. Bunun için karşılaştırma içinde 4 farklı durumu kontrol ederek sadece şahın hareketini Stockfish'e göndermektedir.

2.4. Robot Kolun Hareketi

Robot kolun kontrolünü ve hareketlerini Arduino üzerinden yapılmaktadır. Arduino Braccio robot kol, 6 adet servo motordan oluşmaktadır [6]. Hareketli nesneler gibi çoklu görevler için çeşitli şekillerde monte edilebilir. Bir kamera veya güneş paneli de eklenebilmektedir. Kolun satranç tahtası üzerindeki karelere ulaşması için her bir servo motorun gerektirdiği açılara ayarlayarak hareketini sağlanamıştır. Bunun için Arduino'daki servo motor derecelerini her bir kare için özel olarak ayarlanmıştır. Program, Arduino'ya serial port üzerinden oynaması gereken hamleyi text olarak göndermektedir. Arduino kısmında ise taş yerleri için tanımlanan fonksiyonları aldığımız text değerleri ile çağırılmıştır.

Tablo 1. Konektör Pin bağlantıları

Bağlayıcı Adı	Kalkan pini	Bağlayıcı Adı	Kalkan pini
M1	11	I0	A0 (14)
M2	10	I1	A1 (15)
M3	9	I2	A2 (16)
M4	6	I3	A3 (17)
M5	5	I4	A4 (18)
M6	3	I5	A5 (19)
TWI	SCL, SDA	Seri	RX0 , TX0

Braccio Shield V4 ile, Braccio'yu ekrana bağlarsanız veya dijital olarak M1, M2, M3, M4, M5, M6 konektörlerini kullanırsa dijital 12 kullanılamamaktadır. Pin 12, soft-start (yumuşak başlangıç) voltaj seviyesini yönetmek için kullanılır. Yumuşak başlangıç işlevi devre dışı bırakılırsa, M1-M6 konnektörleri dijital olarak kullanabilir. M1 ila M6 arasında etiketlenmiş konektörler, Arduino kartının PWM özellikli çıkışlarına bağlanır. I0 ila I5 etiketli konektörler analog girişlerdir. Motor bağlantılarının her birinin koruma sigortası vardır. M1 ila M4, 1.1A ile, M5 ve M6 ise 750mA ile sınırlıdır. Bağlacı ve pin bağlantıları Tablo 1'de gösterilmiştir.



Şekil 4: Robot kol hareketi

TWI konektörü 4-pinli olup Arduino kartınızın Arduino'daki Wire kütüphanesi üzerinden TWI (İki Telli Arabirim) veya I2C (Bütünleşik Devre) protokolünü destekleyen cihazlarla iletişim kurmasını sağlar. SERIAL konektörü 4-pinli olup kartın seri iletişimi destekleyen diğer cihazlarla iletişim kurmasını sağlar. Konektörde toprak ve 5 volt bağlantı sağlanmıştır. Serbestlik dereceleri (degrees of freedom – DOF), anlaşılması gereken çok önemli bir terimdir. Her serbestlik derecesi kol üzerinde bir eklemdir, bükülebileceği ya da dönebileceği ya da çevirebileceği bir yerdir. Robot kolundaki aktüatör sayısına göre, genellikle serbestlik derecelerini belirlenebilmektedir. Arduino Braccio, 6 eksenli bir robot koludur. Altı serbestlik derecesi veya altı eksenli robot olarak adlandırılır, çünkü altı farklı eksende dönebilir. Tablo 2'de, her eksenin işlevlerini belirtilmiştir.

Tablo 2. Eksen Hareketleri

Pozisyon	**İzin Verilen Değerler**
M1: baz dereceleri	0 - 180
M2: omuz dereceleri	15 - 165
M3: dirsek dereceleri	0 - 180
M4: bilek dikey dereceleri	0 - 180
M5: bilek rotasyon dereceleri	0 - 180
M6: kıskaç dereceleri	10 (diş dili açık) – 73 (tutucu kapalı)

2.5 Yüz Tanıma ile Durum Değişimi

Yüz ifadesi tanıma, insan-bilgisayar etkileşimi, örüntü tanıma, görüntü anlama, makine görüşü ve diğer alanlarda yaygın olarak kullanılan, insan duygu tanımanın önemli bir parçasıdır [7]. Onbinden fazla ifade türü vardır ve farklı insanların duygularını ifade etmenin farklı yolları vardır. Yüz sosyal iletişimde önemli bir rol oynar, eşit yüz ifadeleri de hayatidir. Yüz ifadeleri, yalnızca herhangi bir kişinin duyarlılığını veya duygularını ortaya koymakla kalmaz, aynı zamanda zihinsel görüşlerini değerlendirmek için de kullanılabilir. Yüz ifadesi tanıma, birinin yüzündeki ifadeleri tanıma yöntemidir. Mutlu, üzgün, korku, iğrenme, öfkeli, tarafsız, sürpriz gibi ifadeleri tespit etmek için çok çeşitli teknikler önerilmiştir, ancak başkalarının uygulanması zordur. Yüz ifadesi tanıma üç ana adımdan oluşur: (1) Görüntünün yüz algılama ve ön işleme, (2) Özellik çıkarma ve (3) İfade sınıflandırma. Genellikle yüz kemikler, yüz kasları ve cilt dokularından oluşan bir birlikteliktir. Bu kaslar kasıldığında, bükülmüş yüz özellikleri üretilir. Yüz ifadeleri, her türlü bilgiyi iletirken en hızlı iletişim aracıdır. Bu alan, özellikle yapay zeka alanında önemli araştıma konularından biridir. Tensorflow çatısıda bu alanda araştırma yapmak için bir altyapı sunmaktadır [8].

TensorFlow, Google tarafından geliştirilen, evrimsel sinir ağını (CNN), tekrarlayan sinir ağını (RNN) ve diğer derinlik sinir ağı modelini destekleyen ikinci nesil yapay zeka araştırma ve geliştirme sistemidir [9]. Bu sistem Google'ın ürün ve hizmetlerinde yaygın olarak kullanılmaktadır, bir düzineden fazla konuşma tanıma, bilgisayarlı görü, robotik, bilgi alma, bilgi edinme, doğal dil işleme, ilaç alanını içeren 100'den fazla makine öğrenim projesinde kullanılmaktadır. Uygulamada, yüz tanıma işlemi için TensorFlow ile yazılmış açık kaynak olan kodu kullanarak yüzdeki duygu değişimlerini(mutlu,kızgın,doğal…vb.) alınmasını sağlayan "face-api" kullanılmıştır [10]. Daha sonra elde edilen duygu değerlerine göre Stockfish'te bulunan duygu değeri (contemptValue) değerini 0 ile 100 arasında duyguya bağlı olarak ayarlanmıştır.

Tablo 2. Duygu Değerleri

Duygun	Duygu Değeri
Üzgün	0 - 20
Korkulu	20 - 40
Doğal (normal)	40 - 60
Kızgın	60 - 80
Mutlu	80 -100
Tespit Edilemeyen	0 - 100

Duygu değeri 0 ise rakip ile dalga geçmeden oynar,100 değeri ise Stockfish'in oynayış tarzının dalga geçme faktörünün en yüksek olduğu değerdir. Bu duygu değişiminin ölçülmesi için rakibin yüzü gerçek zamanlı olarak izlenir ve her 10 saniyede bir rakibin duygu bilgisi alınarak Stockfish'e gönderilir.

3. Sonuç ve Öneriler

Robotik oyunlar, insan-robot etkileşimi içerisinde önemli bir yer tutmaktadır. Bu çalışmada, çeşitli açık kaynak yazılımlar kullanılarak, doğal bir şekilde insan rakibinin yüz ifadesine göre satranç oynayabilen bir robotik sistemi sunulmuştur. Uygulamanın ileriki safhalarında, mobil ve internet üzerinden kullanım ile sesli komut algılama özellikleri eklenmesi planlanmaktadır. Geleceğin, insan ve robotların daha yakından ve sık etkileşimde bulunacağı bir robot uygarlığı dönemi olduğunu söylemek yanlış olmaz. Yapay olarak zeki robotların askeri, hükümet, iş dünyası, eğitim merkezlerinde önemli bir yer bulacağını öngörülmektedir.

4. Teşekkür ve Notlar

Bu uygulama Bolu Abant İzzet Baysal Üniversitesi Bilimsel Araştırmalar Projesi (BAP) 2018.09.01.1372 nolu proje kapsamında gerçekleştirilmiştir. Projeye destek veren Bilgisayar Mühendisliği Bölümü öğrencileri Fatih Tıngır, Büşra Şeker ve Cem Eren Sert'e teşekkürlerimi sunarım. Geliştirilen robot kol, 7 Mayıs 2019 tarihinde Çankaya Üniversitesi'nde onuncusu gerçekleştirilen RoboÇankaya'19 Robot Yarışması'nda serbest kategoride birinci ve 24-26 Nisan 2019 tarihleri arasında Marmara Üniversitesi'nde dördüncüsü düzenlenen "Marmara Mekatronik ve İnovasyon Günleri-MİG" Robot Yarışması'nda ise aynı kategoride ikinci olmuştur.

Kaynakça

[1] Öztürk, M. (2014). *Antropomorfik Robotların Dinamiği Ve Adaptif Kontrol Uygulamaları: Matlab/simulink Modelleme* (Doctoral dissertation, Fen Bilimleri Enstitüsü)

[2] Capek, K. (2004). *RUR (Rossum's universal robots)*. Penguin.

[3] Standage, T. (2002). *The mechanical Turk: the true story of the chess-playing machine that fooled the world*. Allen Lane The Penguin Press.

[4] Bradski, G., Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, Inc.

[5] GNU General Public License Version 3, 2007, [online] Available: http://www.gnu.org/licenses/gpl-3.0.de.html

[6] Tinkerkit Braccio Robot. https://store.arduino.cc/tinkerkit-braccio

[7] Yazar, I., Yavuz, H. S., Çay, M. A. (2009). Temel Bileşen Analizi Yönteminin ve Bazı Klasik ve Robust Uyarlamalarının Yüz Tanıma Uygulamaları. *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, *22*(1), 49-63.

[8] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (pp. 265-283).

[9] Kayikci, S. (2018, September). A Deep Learning Method for Passing Completely Automated Public Turing Test. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 41-44). IEEE.

[10] Indrawan, P., Budiyatno, S., Ridho, N. M., & Sari, R. F. (2013). Face recognition for social media with mobile cloud computing. *International Journal on Cloud Computing: Services and Architecture*, *3*(1), 23-35.