



Teknoloji Fakültesi

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

### GÖRÜNTÜ İŞLEME TEKNİĞİ İLE SATRANÇ OYUN HAMLESİ TESPİTİ

"

#### BİTİRME PROJESİ ARA RAPORU

Bilgisayar Mühendisliği Bölümü

**DANIŞMAN**

Doç. Dr. ÖNDER DEMİR

İSTANBUL, 2025

"



Teknoloji Fakültesi

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

### GÖRÜNTÜ İŞLEME TEKNİĞİ İLE SATRANÇ OYUN HAMLESİ TESPİTİ

"

#### BİTİRME PROJESİ

Bilgisayar Mühendisliği Bölümü

#### DANIŞMAN

Doç. Dr. ÖNDER DEMİR

İSTANBUL, 2025

"

**MARMARA ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Emine Yiğit , Aslı Cennet Ercan , Yusuf Doğan tarafından **GÖRÜNTÜ İŞLEME TEKNİĞİ İLE SATRANÇ OYUN HAMLESİ TESPİTİ** başlıklı proje çalışması, xxx tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Dr. Öğr. Üyesi xxx xxx  
Marmara Üniversitesi  
Prof. Dr. Xxx xxx  
Marmara Üniversitesi  
Prof. Dr. Xxx xxx  
Marmara Üniversitesi

**(Danışman)**

(Üye)

(Üye)

(İMZA).....

(İMZA).....

(İMZA).....

## ÖNSÖZ

Proje çalışmamız süresince karşılaştığım bütün problemlerde, sabırla yardım ve bilgilerini esirgemeyen, tüm desteğini sonuna kadar yanımda hissettiğim değerli hocalarım, sayın Dr. Öğr. Üyesi Xxx xxx ve sayın Prof. Dr. Xxx xxx' a en içten teşekkürlerimi sunarım.

Bu proje çalışması fikrinin oluşması ve ortaya çıkmasındaki önerisi ve desteğinden dolayı değerli hocam Dr. Öğr. Üyesi Xxx xxx' a teşekkür ederim.

Proje çalışmam sırasında maddi ve manevi desteklerini esirgemeyen okul içerisinde ve okul dışında her zaman yanımda olan değerli çalışma arkadaşlarım ve hocalarım Doç. Dr. Xxx xxx ve Dr. Öğr. Üyesi ' xxx xxx a sonsuz teşekkürlerimi sunarım.

# İÇİNDEKİLER

## 1. GİRİŞ

- 1.1. Proje Çalışmasının Amacı ve Önemi
- 1.2. Satranç Oyunu

## 2. MATERYAL VE METOT

- 2.1. OpenCV Kullanımı
- 2.2. YOLO Kullanımı
- 2.3. Optical Flow ile Hareket Algılama
- 2.4. Histogram of Gradients (HOG) Kullanımı
- 2.5. Stockfish ve Python-Chess Kullanımı
- 2.6. Makine Öğrenmesi
- 2.7. Yapay Zeka
- 2.8. Görüntü İşleme

## 3. YÖNTEM VE UYGULAMA

- 3.1. Görüntü Ön İşleme
- 3.2. Satranç Tahtasının Algılanması
- 3.3. Perspektif Düzeltme ve Tahtanın Normalizasyonu
- 3.4. Karelerin Tespiti ve Etiketlenmesi
- 3.5. Satranç Taşlarının Tespiti
- 3.6. Hamle Algılama ve Takip
- 3.7. Hamle Geçerliliği ve Satranç Kurallarına Uyumluluk Kontrolü
- 3.8. Sonuçların Görselleştirilmesi ve Kayıt Altına Alınması

## 4. YAPILACAK ÇALIŞMALAR

- Satranç tahtasının OpenCV ile tespit edilmesi ve 8x8 karelere ayrılması
- YOLOv8 modelinin taşları tanıyacak şekilde eğitilmesi ve test edilmesi
- Hamlelerin Optical Flow yöntemiyle takip edilmesi
- Python-chess kütüphanesi ile yapılan hamlelerin doğrulanması
- Verilerin PGN formatında kaydedilerek bir oyun kaydı oluşturulması

## 5. SONUÇLAR

## 6. KAYNAKLAR

### ÖZET

#### GÖRÜNTÜ İŞLEME TEKNİĞİ İLE SATRANÇ OYUN HAMLESİ TESPİTİ

Bu çalışma, bilgisayarla görme ve yapay zeka teknolojilerini kullanarak satranç oyunlarını otomatik olarak analiz eden bir sistemin geliştirilmesini amaçlamaktadır. Geleneksel satranç oyunlarında hamlelerin manuel olarak kaydedilmesi zaman alıcı ve hata yapmaya açık bir süreçtir. Bu nedenle, kamera tabanlı bir sistem ile hamlelerin gerçek zamanlı olarak algılanması, yanlış hamlelerin tespit edilmesi ve otomatik notasyon kaydı oluşturulması hedeflenmiştir.

Projede, satranç tahtasının ve taşlarının tespiti için **OpenCV**, nesne tanıma işlemleri için **YOLOv8**, taş hareketlerinin izlenmesi için **Optical Flow**, hamlelerin kurallara uygunluğunu değerlendirmek için **Stockfish satranç motoru ve Python-chess kütüphanesi** kullanılmıştır. Görüntü işleme teknikleri ile satranç tahtasının sınırları belirlenmiş, perspektif düzeltme işlemleri uygulanmış ve taşların hareketleri takip edilerek hamleler analiz edilmiştir.

Önerilen sistem, satranç oyuncularına manuel kayıt ihtiyacını ortadan kaldırarak daha hızlı ve hatasız bir oyun deneyimi sunmaktadır. Ayrıca, hamlelerin otomatik olarak analiz edilmesi, oyuncuların stratejilerini geliştirmelerine yardımcı olmaktadır. Geliştirilen modelin, amatör ve profesyonel satranç oyuncuları, turnuva organizatörleri ve eğitimciler için önemli bir katkı sağlaması beklenmektedir.

**Anahtar Kelimeler:** Bilgisayarla Görme, Satranç, Yapay Zeka, OpenCV, YOLOv8, Optical Flow, Python-chess, Stockfish

Mart, 2025

Emine Yiğit, Ashı Cennet Ercan ,Yusuf Doğan

## **ABSTRACT**

### **DETECTION OF CHESS MOVES USING IMAGE PROCESSING TECHNIQUES**

In this study, a system is developed to automatically detect and analyze chess moves using computer vision and artificial intelligence technologies. Traditional chess games require manual move notation, which is time-consuming and prone to human errors. To address this issue, a camera-based system is designed to detect chessboard positions in real-time, identify the movements of pieces, and validate moves according to chess rules.

The proposed system employs OpenCV for chessboard and piece detection, YOLOv8 for object recognition, Optical Flow for movement tracking, and Stockfish and Python-Chess libraries for move validation. Image processing techniques such as Canny Edge Detection, Perspective Transformation, and Contour Detection are utilized to extract chessboard structures. The system also records moves in Portable Game Notation (PGN) format, allowing users to analyze their games later.

By automating the process of move detection and validation, this system provides an efficient and error-free experience for chess players, tournament organizers, and trainers. The integration of artificial intelligence and computer vision enhances strategic gameplay analysis, making chess more accessible and interactive for players at all levels.

**Keywords:** Computer Vision, Chess, Artificial Intelligence, OpenCV, YOLOv8, Optical Flow, Python-Chess, Stockfish

**March, 2025**

**Emine Yiğit, Aslı Cennet Ercan ,Yusuf Doğan**

## KISALTMALAR

1. **PGN** – Portable Game Notation (Taşınan Oyun Notasyonu)
2. **RGB** – Red Green Blue (Kırmızı, Yeşil, Mavi – Renk Uzayı)
3. **HOG** – Histogram of Oriented Gradients (Yönlendirilmiş Gradyan Histogramı)
4. **CVPR** – IEEE Conference on Computer Vision and Pattern Recognition
5. **YOLO** – You Only Look Once (Derin öğrenme tabanlı nesne tespiti algoritması)
6. **AI** – Artificial Intelligence (Yapay Zeka)
7. **GPO** – Gaussian Blur (Görüntü işleme filtresi)
8. **CNN** – Convolutional Neural Networks (Evrensel Sinir Ağları)
9. **INS** – Inertial Navigation System (Ataletsel Navigasyon Sistemi)



## ŞEKİL LİSTESİ

- Resim-1
- Resim-2

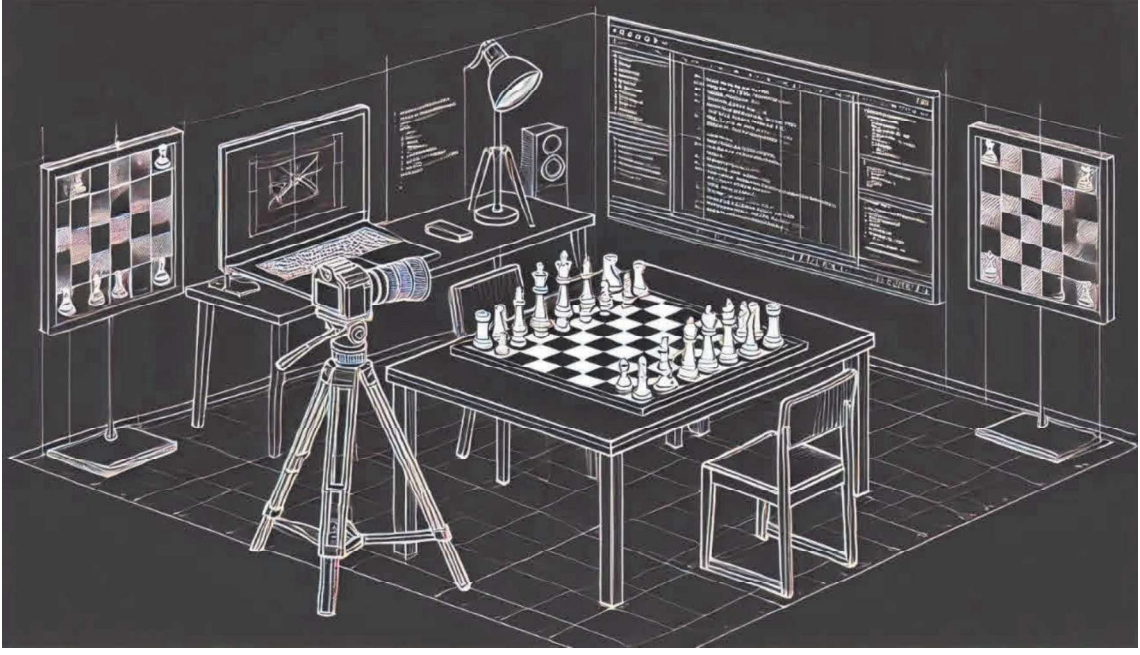
## 1. GİRİŞ

Satranç, strateji ve mantık üzerine kurulu, dünya çapında milyonlarca insanın oynadığı ve zeka gelişimini destekleyen bir oyun olarak büyük bir öneme sahiptir. Özellikle dijitalleşen dünyada, satranç oyuncularını ve turnuva organizatörlerini için teknolojinin getirdiği yeniliklerin kullanımı giderek artmaktadır. Geleneksel satranç oyunlarında hamlelerin manuel olarak kaydedilmesi, zaman alıcı ve hata yapmaya açık bir süreçtir. Ayrıca, oyun sırasında yapılan yanlış hamlelerin fark edilmesi bazen zor olabilmektedir. Bu bağlamda, görüntü işleme ve yapay zeka teknolojilerinin sunduğu imkanlarla satranç oyunlarını otomatik olarak takip etmek, oyunculara daha hızlı ve verimli bir deneyim sunma potansiyeli taşımaktadır.

Bu projede, kamera yardımıyla satranç tahtasında yapılan hamlelerin gerçek zamanlı olarak izlenmesi ve yapay zeka desteği ile yanlış hamlelerin tespit edilmesi hedeflenmektedir. Projenin bir diğer önemli unsuru ise, oyuncuların yaptığı hamlelerin otomatik notasyon ile kaydedilmesi ve böylece manuel kaydetme ihtiyacının ortadan kaldırılmasıdır. Yapay zeka ve görüntü işleme tekniklerinin entegre edilmesiyle oluşturulacak bu sistem, oyuncuların stratejiye daha fazla odaklanmasını ve oyun analizlerini daha kolay gerçekleştirmesini sağlayacaktır.

Projenin hem amatör hem de profesyonel satranç oyuncularını için önemli faydalar sunması amaçlanmaktadır. Oyun sırasında yapılan yanlış hamlelerin anında fark edilmesi, oyuncuların hatalarını zamanında düzeltmelerine ve oyunu kurallara uygun bir şekilde sürdürebilmelerine yardımcı olacaktır. Ayrıca otomatik notasyon, satranç turnuvalarında ve eğitim süreçlerinde büyük kolaylık sağlayarak oyunların detaylı bir şekilde analiz edilmesine olanak tanıyacaktır.

Görüntü işleme ve yapay zeka teknolojileri sayesinde elde edilecek bu yenilikçi yaklaşım, satranç oyunlarını daha dinamik ve hatasız bir şekilde oynamayı mümkün kılarken, hem bireysel oyunculara hem de satranç topluluğuna önemli katkılar sunacaktır. Bu proje, satranç oyunlarına teknolojik bir çözüm getirerek, oyunun oynanışını ve analiz süreçlerini dönüştürme potansiyeline sahiptir.



Resim-1

### 1.1. Proje Çalışmasının Amacı ve Önemi

Bu projenin amacı, görüntü işleme ve yapay zeka teknolojileri kullanarak satranç oyunlarında kamera yardımıyla yapılan hamlelerin otomatik olarak algılanması, yanlış hamlelerin tespit edilmesi ve oyuncuların oynadığı taşların notasyonlarının otomatik olarak kaydedilmesidir. Geleneksel satranç oyunlarında hamlelerin manuel olarak kaydedilmesi oldukça zaman alıcı olup hata riskini de artırmaktadır. Proje, bu süreci dijitalleştirip otomatik hale getirerek oyunculara daha hızlı, doğru ve verimli bir deneyim sunmayı hedeflemektedir. Ayrıca yanlış hamlelerin anında tespit edilip oyunculara geri bildirim verilmesi, oyun kurallarına uygun hamle yapılmasını sağlar ve oyuncuların hatalarını anında fark etmelerine olanak tanır.

Bu proje, satranç oyuncularının manuel olarak hamle kaydetme gerekliliğini ortadan kaldırarak, oyun sırasında daha fazla stratejiye odaklanmalarına imkan tanır. Otomatik notasyon özelliği, oyuncuların hamlelerini doğru ve eksiksiz bir şekilde kaydederken, aynı zamanda oyun sonrasında yapılan hamlelerin analiz edilmesine ve stratejilerin geliştirilmesine katkı sağlar. Ayrıca bu sistem, profesyonel satranç turnuvalarında maçların hızlı ve güvenilir bir şekilde kayıt altına alınmasını mümkün kılar.

Projenin sağlayacağı faydalardan biri, oyuncuların hamlelerini kaydetme sürecini

kolaylaştırırken zaman ve doğruluk açısından büyük avantaj sunmasıdır. Otomatik hamle tespiti ve notasyon sayesinde, insan kaynaklı hatalar ve zaman kayıpları ortadan kalkar. Yapay zeka destekli analizler, oyuncuların hamlelerini gözden geçirip performanslarını geliştirmelerine yardımcı olur. Ek olarak, oyun sırasında yanlış hamlelerin hızlıca algılanıp oyuncuların uyarılması, satranç oyunlarının kurallarına uygun bir şekilde ilerlemesine ve daha adil bir oyun deneyimi sunulmasına katkıda bulunur.

Sonuç olarak, bu proje hem amatör hem de profesyonel satranç oyuncularının oyun deneyimlerini iyileştirerek, oyunun hem eğitim hem de performans analiz süreçlerinde önemli bir kolaylık sağlayacaktır. Görüntü işleme ve yapay zeka ile entegre edilen bu teknoloji, satranç dünyasında yenilikçi bir adım olarak öne çıkmaktadır.

## 1.2. Satranç Oyunu

Satranç, tarihi çok eski zamanlara dayanan, iki oyuncu arasında oynanan bir zekâ oyunudur. Oyun, 8x8 boyutlarında toplam 64 kareden oluşan bir alan üzerinde 32 adet taş ile oynanır. Satranç tahtası karelerinin yarısı koyu (siyah), diğer yarısı açık (beyaz) renklerden oluşur. Oyun başladığında 16 siyah ve 16 beyaz taş bulunmaktadır. Bu taşlar; her bir oyuncu için 1 adet şah, 1 adet vezir, 2 adet kale, 2 adet fil, 2 at ve 8 tane piyondan meydana gelir. Oyuncular arasında bir oyuncu beyaz ve diğer oyuncu siyah taşları oynayacak şekilde renk seçimi yapılır ve oyuna beyaz taşları seçen oyuncu başlar. Her oyuncunun bir seferde bir hamle yapmasıyla oyun gelişir. Oyunun amacı karşı tarafın şahını ele geçirmektir. Bir başka deyişle, şahın bulunduğu karenin tehdit altına alınması ve bu tehdidi engelleyecek karşı hamlenin olanaksız hale gelmesi durumudur (Satranç Dünyası, 2017). Her taşın farklı bir hareket kabiliyeti bulunmaktadır. Bu hareketler aşağıda özetlenmiştir:

- Piyon: Bir kare ileri gidebilir. Yalnızca ilk defa hareket ettiğinde iki kare ilerleyebilir. İleri doğru giderken sol ya da sağ çaprazındaki taşları ele geçirebilir. Piyon, geçerken alma (en passant) adı verilen özel bir hamle de yapabilir. Bu durum, hiç hamle yapmamış olan piyonun iki kare öne çıkarak rakip piyon ile yan yana gelmesi sonucu rakip piyon tarafından ele geçirilmesidir.
- At: “L” harfine benzer şekilde yukarı, aşağı, sağa ve sola doğru hareket edebilir. At, gittiği konumdaki rakip taşı ele geçirebilir. Diğer taşlardan farklı olarak başka bir taşın üzerinden atlayarak hareket edebilir.

- Kale: Yukarı, aşağı, sağa ve sola istenilen kare sayısı kadar hareket edebilir. Gittiği konumdaki rakip taşı ele geçirebilir.
- Fil: Çapraz olarak istenilen kare sayısı kadar hareket edebilir. Gittiği konumdaki rakip taşı ele geçirebilir. Bir oyuncunun sahip olduğu iki filin biri sadece beyaz, diğeri ise sadece siyah kareler üzerinde hareket edebilir.
- Vezir: Hem kale gibi yukarı, aşağı, sağa ve sola, hem de fil gibi çapraz olarak istenilen kare sayısı kadar hareket edebilmektedir. Gittiği konumdaki rakip taşı ele geçirebilir.
- Şah: Her yöne sadece bir kare ilerleyerek hareket edebilir. Gittiği konumdaki rakip taşı ele geçirebilir. Şah, kendisine tehdit oluşturabilecek rakip taşların hareket alanlarına giremez (Satranç Dünyası, 2017).

## **2. MATERYAL VE METOT**

Bu projede OpenCV, YOLOv8, Optical Flow ve python-chess gibi kütüphaneler kullanılarak satranç tahtası ve taşların tespiti gerçekleştirilecektir.

### **2.1.OpenCV Kullanımı**

OpenCV, görüntü işleme ve nesne tanıma işlemlerinde kullanılan güçlü bir açık kaynak kütüphanesidir. Bu projede, tahta algılama, taşların tespiti ve hamleleri izleme için OpenCV'den faydalanılacaktır.Canny Edge Detection ile satranç tahtasının sınırları belirlenecek ve 8x8 karelere bölünecektir.

Perspective Transform ile kamera açısına göre tahtanın görüntüsü düzeltilerek analiz edilebilir hale getirilecektir.Template Matching ve Contour Detection ile taşların konumu belirlenecektir.

### **2.2.YOLO Kullanımı**

YOLO (You Only Look Once), nesne tespiti için kullanılan derin öğrenme tabanlı bir kütüphanedir.Satranç taşlarını farklı açılardan ve ışık koşullarında tanıyabilmek için YOLOv8 modeli eğitilecektir.Model, piyon, at, fil, kale, vezir ve şah gibi taş türlerini doğru bir şekilde tanımlayacak şekilde optimize edilecektir.

### **2.3.Optical Flow ile Hareket Algılama**

Optical Flow, bir nesnenin bir çerçeveden diğerine nasıl hareket ettiğini belirlemek için kullanılan bir yöntemdir.

Satranç taşlarının hareketlerini belirlemek için Lucas-Kanade Optical Flow algoritması uygulanacaktır. Taşın eski ve yeni konumu tespit edilerek hamlenin hangi taş tarafından yapıldığı anlaşılabilecektir.

### **2.4.Histogram of Gradients (HOG) Kullanımı**

HOG, görüntüdeki kenar ve doku özelliklerini belirlemek için kullanılan bir yöntemdir. Satranç taşlarının kenarlarını belirlemek ve sınıflandırmak için HOG özellik çıkarımı yapılacaktır. Taşların hareketlerini belirlemek için HOG ile elde edilen değerler karşılaştırılarak değişiklikler analiz edilecektir.

### **2.5.Stockfish ve Python-Chess Kullanımı**

Stockfish, dünyanın en güçlü açık kaynak satranç motorlarından biridir. Python-chess ise satranç kurallarını uygulamak için kullanılan bir Python kütüphanesidir.

Yapılan hamlelerin kurallara uygun olup olmadığını belirlemek için Stockfish motoru ile analiz yapılacaktır. Geçersiz hamleler tespit edilerek kullanıcıya uyarı verilecektir. Tüm hamleler PGN formatında kaydedilerek oyunun analizi sağlanacaktır.

### **2.6.Makine Öğrenmesi**

Makine öğrenmesi, verilerden örüntüler çıkararak sistemlerin kendiliğinden öğrenmesini sağlayan bir yapay zeka alt alanıdır. Bu projede, taşları tanıma ve hareketlerini algılama için makine öğrenmesi kullanılacaktır.

### **2.7.Yapay Zeka**

Yapay zeka, bilgisayar sistemlerinin insan benzeri zekasal işlevleri yerine getirmesini sağlayan bir teknolojidir. Bu projede, hamlelerin kurallara uygunluğunu denetlemek için yapay zeka tabanlı bir analiz kullanılacaktır.

### **2.8.Görüntü İşleme**

Görüntü işleme, dijital görüntüler üzerinde analiz ve dönüşüm yapma işlemlerini kapsar. Taşların hareketlerini algılamak ve kaydetmek için OpenCV gibi görüntü işleme kütüphaneleri kullanılacaktır.





Resim-2

### 3.YÖNTEM VE UYGULAMA

Bu çalışmada, satranç tahtası üzerindeki hamleleri otomatik olarak tespit edebilmek için bilgisayarla görme tekniklerinden faydalanılmıştır. İlk yöntem, çeşitli ön işleme adımları ile başlayıp, satranç tahtasının belirlenmesi, perspektif düzeltme, karelerin tespiti, taşların sınıflandırılması ve hamlelerin analiz edilmesi süreçlerini içermektedir. Kullanılan algoritmalar, görüntü işleme ve yapay zeka tekniklerinin entegrasyonu ile hassasiyet ve doğruluk oranlarını artırmayı hedeflemektedir.

#### 3.1. Görüntü Ön İşleme

Satranç tahtasının algılanması ve hamlelerin doğrulukla tespit edilebilmesi için, görüntü ön işleme adımları kritik bir rol oynamaktadır. Bu adımlar, görüntüdeki gürültünün azaltılması, kenarların belirginleştirilmesi ve taşların tespitinin kolaylaştırılması amacıyla uygulanmaktadır.

İlk olarak, giriş görüntüsü RGB renk uzayından gri tonlamalı (grayscale) formata dönüştürülmüştür. Renk bilgisinin tahtanın konumunu belirleme sürecinde gereksiz olduğu göz önüne alınarak, gri tonlamaya çevirme işlemi ile hesaplama yükü azaltılmıştır. Daha sonra, düşük frekanslı gürültülerin etkisini azaltmak amacıyla “**Gaussian Blur filtresi**” uygulanmıştır. Bu işlem, görüntüdeki keskin geçişleri yumuşatarak, kenar tespit

algoritmalarının daha kararlı çalışmasını sağlamaktadır.

Son olarak, görüntüdeki kenarları belirleyebilmek amacıyla “**Canny Edge Detection**” yöntemi kullanılmıştır. “Canny” algoritması, öncelikle görüntü üzerinde “Sobel” operatörleri ile gradyan hesaplaması yapmakta ve ardından belirli eşik değerleri ile kenar bölgelerini belirlemektedir. Bu işlem, satranç tahtasının dış çerçevesinin ve iç karelerin daha belirgin hale getirilmesini sağlamaktadır.

### 3.2. Satranç Tahtasının Algılanması

Ön işleme aşamasından sonra, görüntüdeki dörtgen yapıların belirlenmesi için **kontur tespiti (contour detection)** yöntemi uygulanmıştır. OpenCV kütüphanesinde yer alan `cv2.findContours` fonksiyonu, görüntüdeki nesnelerin dış sınırlarını belirleyerek, dörtgen biçimli yapıları tespit etmeye yardımcı olmaktadır.

Algılanan konturlar arasından en büyük dörtgen yapı seçilmiş ve bunun satranç tahtası olup olmadığı **hiyerarşik analiz** ile doğrulanmıştır. Dörtgenin belirlenen köşe noktaları, **Harris köşe tespiti algoritması** ile kontrol edilerek, satranç tahtasına ait olup olmadığı değerlendirilmiştir.

### 3.3. Perspektif Düzeltme ve Tahtanın Normalizasyonu

Satranç tahtasının görüntüde farklı açılardan alınmış olması, taşların tespitinde hata oranını artırabileceğinden, perspektif bozukluklarının düzeltilmesi gerekmektedir. Bu amaçla, **homografi dönüşümü** uygulanarak satranç tahtası standart bir referans çerçevesine oturtulmuştur.

Perspektif düzeltme için aşağıdaki adımlar uygulanmıştır:

1. **Dörtgenin köşe noktalarının sıralanması:** `cv2.approxPolyDP` algoritması kullanılarak dört köşe belirlenmiştir.
2. **Kaynak ve hedef noktalarının eşleştirilmesi:** Köşe noktaları, standart  $800 \times 800$  piksel boyutundaki bir kareye oturtulmuştur.
3. **Homografi matrisinin hesaplanması:** `cv2.getPerspectiveTransform` fonksiyonu ile dönüşüm matrisi elde edilmiştir.
4. **Görüntü dönüştürme işlemi:** `cv2.warpPerspective` fonksiyonu ile perspektif bozukluğu düzeltilmiştir.

Bu işlemin sonucunda, satranç tahtasının tüm kareleri eşit büyüklükte ve standart formata uygun hale getirilmiştir.

### 3.4. Karelerin Tespiti ve Etiketlenmesi

Perspektif düzeltme işleminin ardından, satranç tahtasının  $8 \times 8$  kareye bölünmesi ve her bir karenin adlandırılması gerçekleştirilmiştir. Uluslararası satranç notasyonu kullanılarak her kare (a1–h8) formatında etiketlenmiştir. Karelerin doğru şekilde ayrıldığını doğrulamak için, her karenin merkezi belirlenmiş ve satranç tahtasının grid yapısı çıkarılmıştır.



### 3.5. Satranç Taşlarının Tespiti

Satranç taşlarının tespitinde, farklı bilgisayarla görme teknikleri bir arada kullanılmıştır.

1. **Arka plan çıkarma (background subtraction)** yöntemi kullanılarak tahtada meydana gelen değişiklikler analiz edilmiştir.
2. **Renk tabanlı segmentasyon** tekniği ile taşların sınıflandırılması gerçekleştirilmiştir. HSV renk uzayında yapılan analizle, beyaz ve siyah taşlar arasındaki parlaklık farkı kullanılarak taşların hangi karelerde bulunduğu belirlenmiştir.
3. **Özellik tabanlı nesne tanıma** yaklaşımı ile taşların şekilleri analiz edilmiş ve taş tipleri belirlenmiştir.

Bu süreçte, cv2.threshold ve cv2.inRange fonksiyonları kullanılarak renk eşikleme işlemi gerçekleştirilmiş, ardından cv2.findContours fonksiyonu ile taş konturları belirlenmiştir.

### 3.6. Hamle Algılama ve Takip

Hamlelerin belirlenmesi için iki ardışık görüntü karşılaştırılmış ve değişiklikler tespit edilmiştir.

1. İlk olarak, satranç tahtasının başlangıç durumu kaydedilmiştir.
2. Ardından, her yeni karede tahtanın mevcut durumu analiz edilerek değişiklikler kaydedilmiştir.
3. cv2.absdiff fonksiyonu ile iki görüntü arasındaki farklar hesaplanarak hareket eden taş tespit edilmiştir.
4. Taşın başlangıç ve bitiş kareleri belirlenmiş ve hamle e2-e4 formatında kaydedilmiştir.

### 3.7. Hamle Geçerliliği ve Satranç Kurallarına Uyumluluk Kontrolü

Tespit edilen hamlelerin geçerli olup olmadığını belirlemek için satranç kuralları ile karşılaştırma yapılmıştır.

- Hareket eden taşın kurallara uygun olup olmadığı kontrol edilmiştir.
- Özel hamleler (rok, en passant, piyon terfisi) değerlendirilmiştir.
- Şah tehdidi olup olmadığı analiz edilerek illegal hamleler raporlanmıştır.

Bu işlemler, satranç motoru kullanılmaksızın kodlanmış basit kural tabanlı bir sistem ile gerçekleştirilmiştir.

### 3.8. Sonuçların Görselleştirilmesi ve Kayıt Altına Alınması

Tespit edilen hamleler görselleştirilmiş ve sistem tarafından kaydedilmiştir.

- Satranç tahtasının anlık durumu bir grafik arayüzde gösterilmiştir.
- Algılanan hamleler bir veri tabanına kaydedilmiştir.
- Hamlelerin geri alınabilmesi için zaman damgalı kayıtlar tutulmuştur.

#### 4. YAPILACAK ÇALIŞMALAR

- Satranç tahtasının OpenCV ile tespit edilmesi ve 8x8 karelere ayrılması,
- YOLOv8 modelinin taşları tanıyacak şekilde eğitilmesi ve test edilmesi,
- Hamlelerin Optical Flow yöntemiyle takip edilmesi,
- Python-chess kütüphanesi ile yapılan hamlelerin doğrulanması,
- Verilerin PGN formatında kaydedilerek bir oyun kaydı oluşturulması.

#### 3. SONUÇLAR

Yapılacak hedeflerin çıktı sonuçları değerlendirilecektir.

#### KAYNAKLAR

- [1] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). “You Only Look Once: Unified, Real-Time Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.
- [2] Lucas, B. D., Kanade, T. (1981). “An Iterative Image Registration Technique with an Application to Stereo Vision,” Proceedings of Imaging Understanding Workshop, pp. 121-130.
- [3] Stockfish Chess Engine, <https://stockfishchess.org/>, Erişim Tarihi: 21.03.2025.
- [4] Bradski, G. (2000). “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools.
- [5] Satranç Dünyası, (2017). FIDE Satranç Kuralları. Erişim Adresi: <http://www.satranc.net/satranc-kurallari> (Erişim Tarihi: 04.04.2017).
- [6] Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, Cilt 24, Sayı 1, 2019