



Python ile Derin Öğrenme Kullanarak Otonom Bir Araba Oluşturmak

Ali Kağan Uyanık

Mart 2024



Özet

Bu projenin amacı, Python programlama dili ve çeşitli derin öğrenme, bilgisayarla görü ve makine öğrenimi teknikleri kullanarak otonom bir araç geliştirmektir. Otonom aracın çevresini algılamak, kararlar almak ve güvenli bir şekilde hareket etmek için gerekli yeteneklere sahip olması hedeflenmektedir.

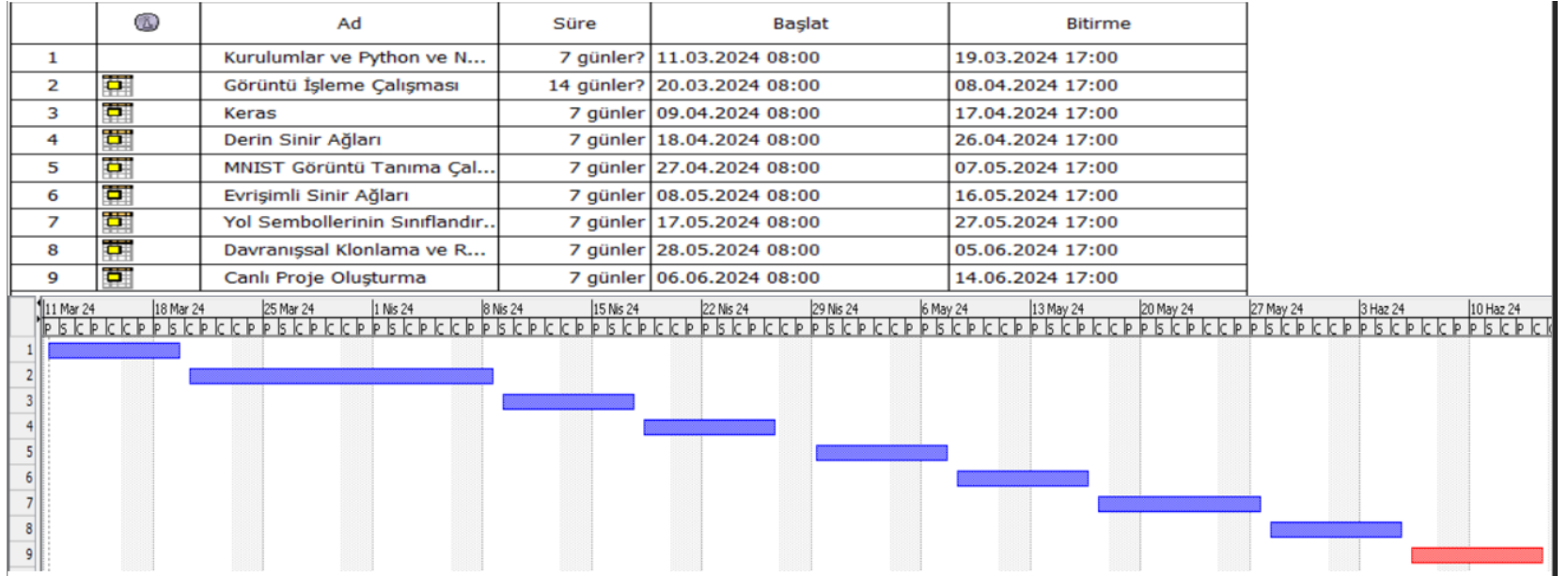
1 Giriş

Teknolojinin hızla gelişmesiyle birlikte, otonom sürüş sistemleri[1]otomotiv sektöründe devrim niteliğinde bir değişim başlatmış durumda. Geleneksel sürüş alışkanlıklarını geride bırakarak, araçlar artık kendi kararlarını alabilir, çevrelerini analiz edebilir ve yolculuk boyunca tüm kontrolü ele alabilir hale geldi. Otonom sürüş, sadece bir teknolojik yenilik olmanın ötesinde, günümüz toplumlarının ve insanlığın bir dizi temel ihtiyacına karşılık verebilecek bir potansiyele sahip.

Otonom sürüş, günümüzün yoğun ve karmaşık trafik koşullarında insan hatalarını en aza indirerek güvenliği artırma potansiyeli taşıyor. Ayrıca, uzun yolculuklarda sürücülerin yorgunluğunu azaltarak ve trafik stresini hafifleterek konforu artırabilir. Tüm bunlar, otonom sürüşün sadece bireylerin günlük hayatını değil, aynı zamanda şehir planlaması, ulaşım sistemleri ve çevresel sürdürülebilirlik gibi geniş bir perspektiften toplumları etkileyebilecek bir dönüşümü temsil ediyor.

İnsanlık, nüfusunun artmasıyla birlikte ulaşım sistemlerine olan talebin de arttığı bir dönemden geçmektedir. Şehirler daha yoğun hale gelmekte, trafik sorunları ve çevresel kirlilik gibi sorunlarla karşı karşıya kalmaktadır. Otonom sürüş teknolojisi, bu zorlukları ele almak için potansiyel bir çözüm olarak ortaya çıkmaktadır. Akıllı sensörler, yapay zeka algoritmaları ve sürekli olarak güncellenen haritalar gibi unsurlar sayesinde otonom araçlar, trafiği daha etkin bir şekilde yönetebilir, kazaları önleyebilir ve karbon emisyonlarını azaltabilir.[2]

Aşağıdaki gantt şeması, projenin başlatılmasından sonraki önümüzdeki hftalarda önerilen konseptin ilerlemesine ilişkin proje hedefini temsil etmektedir



Resim 1: Gantt Chart

2 Proje Aşamaları

2.1 Kurulumlar ve Python-Numpy

2.1.1 Kurulumlar

Anaconda dağıtımının sunduğu çeşitli bilimsel paketleri projemde kullanmaya başlamayı hedefliyorum.Bu paketleri başarılı bir şekilde kurarak, projemin temel altyapısını oluşturmayı düşünüyorum.Ayrıca, bilgisayar görüşü ve davranış klonlaması bölümlerinde kullanacağım metin düzenleyiciyi de kurmam gerekiyor.Bu adım, projenin farklı aşamalarında kullanacağım yazılım araçlarını hazır hale getirmem konusunda önemlidir.

2.1.2 Python

Bu bölümde, Python programlama dilinin temellerini öğreneceğim.Python, genel amaçlı bir dil olduğu için farklı türde programlar oluşturmak için tasarlanmıştır.Bu kısımda Python ile veri tipleri,aritmetik işlemler,değişkenler,metotlar,listeler,Python slice() kullanımı,Python üyelik (membership) operatörler,Python'da Mutable,en çok kullanılan Python metod ve fonksiyonları,Python Tuple,Python'da veri yapıları gibi konular hakkında bilgi edinip örnekler yapacağım.

2.1.3 Numpy

Python'un kendisi, temiz ve okunabilir kod yazmamıza olanak tanır; ancak genel amaçlı bir dil olmasına rağmen, büyük veri setlerinin niceliksel analizini doğrudan yapma yeteneğine sahip değildir.İşte bu yüzden numpy adlı özel bir kütüphaneden faydalanmayı düşünüyorum.Bu bölümde dizi, listeler ve numpy stack() üzerinde çalışmalar yapacağım.

2.2 Görüntü İşleme Çalışması

Bu projenin ana odak noktası derin öğrenme olsa da,projenin birçok bölümünde açık kaynaklı bir bilgisayar görüşü kütüphanesi olan openCV'yi kullanacağım.Eğitim verilerini derin sinir ağına beslemeden önce openCV'yi kullanarak bu verileri ön işlemeyi planlıyorum.Şeritleri tanımlamak için openCV'yi kullanmak,otonom araçların yollarda seyretmesinde temel bir adımdır[3].OpenCV'nin sunduğu bazı temel yetenekler arasında kenar tespiti, renk dönüşümleri, morfolojik işlemler ve nesne algılama gibi işlemler bulunmaktadır. Bu işlevler, şeritleri tanımlama gibi karmaşık görevler için temel oluşturur.Bu bölümde,hem openCV'nin yeteneklerini anlamayı hem de derin öğrenme ile nasıl entegre edilebileceğini öğrenmeyi planlıyorum.



Resim 2: Resimdeki Şeritleri Tanıtmak için Kullanacağım Örnek Görsel

2.3 Keras ve Nöral Ağlar

Nöral ağlar biyolojik nöral ağlardan esinlenmiş ve biz insanların öğrenme şeklini belirsiz bir şekilde taklit etmiştir. En temel nöral ağ formu olan perceptron, bir beyin biyolojik nöronu ile benzerlik gösterir. Perceptron, girdileri giriş düğümleri şeklinde almak ve uygun çıkışı transfer etmek üzere eğitilmiştir. Bu, nöronlarda dendritin elektrik sinyallerini alması ve aksonun uygun sinyali göndermek için birden fazla akson terminaline dallanmasıyla benzerdir. Bu bölümde, bir perceptronun çalışma prensiplerine, ağırlıklara ve gradyan inişi konularına bakacağım. Bir aracın şu anda hangi kısmında olduğuna göre kendi başına sürmesini sağlamak için bir nöral ağ kullanmayı planlıyorum. Bu bölümde, sadece basit bir perceptron temelli ağ oluşturmak için Keras'ı kullanmayı düşünüyorum.

2.4 Derin Sinir Ağları

Karmaşık veri ile uğraşmak için daha karmaşık, derin nöral ağlara ihtiyaç duyuyoruz; çünkü bunlar daha yüksek öğrenme kapasitesine sahiptir. Bunu trafik işaretlerinden görüntü özellikleri çıkarmak için ve aynı zamanda bir otonom araç simülasyonundaki belirli bir yolun kısmından görüntü özellikleri çıkarmak için kullanacağım; böylece uygun direksiyon açısını tahmin etmemize yardımcı olacak.

2.5 MNIST Görüntü Tanıma Çalışması

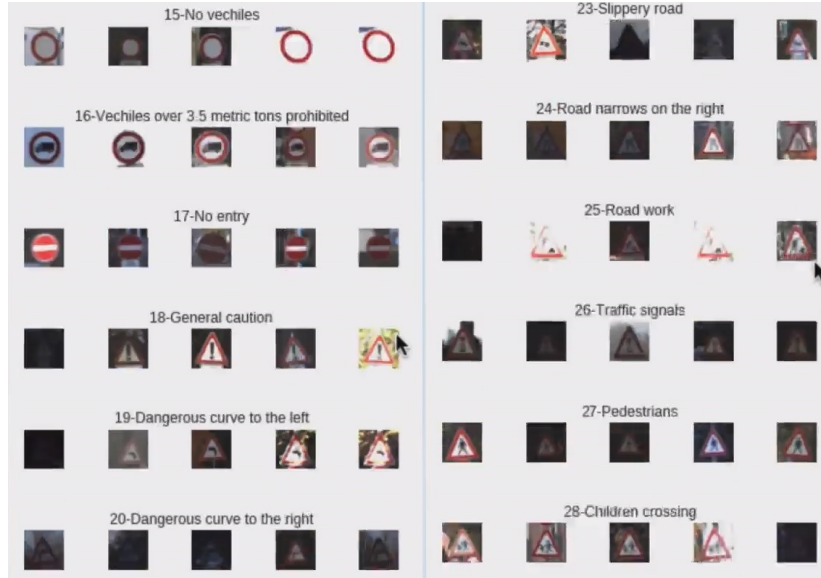
Mnist veri seti ve derin nöral ağları kullanarak bir modelin görüntü verisine uygun olmasını nasıl sağlayacağımı öğrenmeyi planlıyorum..

2.6 Konvolüsyonel Sinir Ağları(CNN)

Konvolüsyonel Sinir Ağları,görüntü sınıflandırma için gidilecek model haline gelmiş, yapay sinir ağlarını geride bırakmıştır.Konvolüsyonel sinir ağlarıyla resimleri nasıl sınıflandıracağımızı öğrenmeyi ve yüksek bir doğruluk oranıyla bu işi yapmayı planlıyorum.

2.7 Yol Sembollerinin Sınıflandırılması

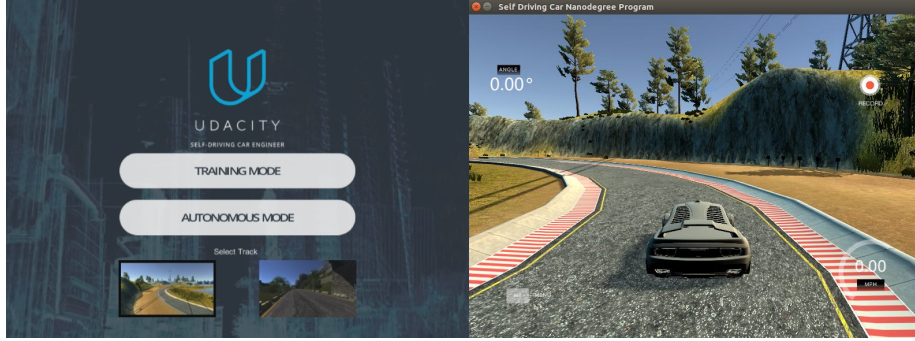
Trafik işaretleri, 43 farklı sınıf arasında sınıflandırılabilen verilerdir.Verit setlerini ön işlemek ve hazırlamak, derin öğrenme alanında son derece önemlidir[4].Şimdiye kadar öğrendiğim her şeyi bir araya getireceğim.



Resim 3: Trafik İşaretleri Veri Örneği

2.8 Davranışsal Klonlama ve Regresyon

Bu bölümde, sürekli bir spektrum temelinde direksiyon açılarını tahmin edebilen bir modeli eğiteceğim. Temel olarak bize Udacity tarafından açık kaynak olarak sağlanan bir otonom araç simülatörünü indireceğim. Simülatörü kullanarak modelim için kendi eğitim verilerimizi oluşturacağım ve simülatör içinde bir aracı eğitim pisti boyunca sürerken sürüş verilerini elde edeceğim. Her sürüş anında bu görüntüleri alacağım. Bu görüntüler, eğitim veri setimi temsil edecek ve her belirli görüntünün etiketi, aracın o belirli anındaki direksiyon açısı olacak. Daha sonra bu görüntüleri konvolüsyonel nöral ağıma gösterecek ve modelin, manuel sürücü olarak bizim davranışımızdan öğrenerek otonom bir şekilde nasıl sürüş yapacağını öğrenmesine izin vereceğim. Bu modelin öğrenmeyi ayarlamayı öğreneceği ana değişken, herhangi bir belirli anındaki aracın direksiyon açısını uygun bir derecede ayarlamayı etkili bir şekilde öğrenecektir. Modelimi eğittikten sonra performansını, arabayı otonom olarak çalıştıracığım tamamen farklı bir test pistinde değerlendireceğim. Arabayı doğru bir şekilde eğitebilirsek, ikinci pistimizde çok iyi bir performans sergileyecek ve kendi kendine sürüş yapacaktır.



Resim 4: Simülatör içi görüntüler

3 Literatür Araştırması

Bu makalede[5], derin öğrenme yöntemlerinin otonom araçlar alanında uygulanmasının teorileri ve uygulamaları üzerine son araştırmaları incelemektedir. Otonom araçlardaki temel sorunlar ve derin öğrenme yöntemlerine dayalı çözümleri, engel tespiti, sahne tanıma, şerit tespiti, navigasyon ve rota planlama gibi konular analiz edilir.

Bu makalede[6], derin öğrenmenin bilgisayar görüşünü devrimlendirdiğini ve otonom araçların yeteneklerinin temel teknolojisi olduğunu belirtmektedir. Konvolüsyonel Sinir Ağları (CNN'ler), nesne tespiti görevini geliştirmek için derin öğrenme devriminin kalbinde bulunmaktadır. CNN'lere dayanan bir dizi başarılı nesne tespit sistemi önerilmiştir.

Bu makalede[7], Udacity'den alınan verilerle eğitilen bir sinir ağı sürümü ve kendi topladıkları (simüle edilmiş) verilerle eğitilen özel sürümlerinin karşılaştırması yapılıyor.

4 Referanslar

GitHub -https://github.com/woges/UDACITY-self-driving-car/tree/master/term1_project3_behavioral_cloning

GitHub -<https://github.com/SakshayMahna/P4-BehavioralCloning>

GitHub -https://github.com/11Source11/How_to_simulate_a_self_driving_car

GitHub -<https://github.com/naokishibuya/car-behavioral-cloning>

GitHub -<https://github.com/udacity/self-driving-car-sim>

Kaggle-<https://www.kaggle.com/datasets/aslanahmedov/self-driving-carbehavioural-cloning?select=IMG>

5 Beklenen Sonular

Proje, derin sinir aęları ve konvolüsyonel sinir aęları kullanarak sürücölük davranışını klonlamak için başarıyla bir model geliştirmiştir.Elde edilen sonuçlar, eğitilen modelin pistte başarıyla sürüş yapabilme yeteneğini deęerlendirmektedir.Modelin pistte ne kadar başarılı olduęu, doęrulama ve test aşamalarında elde edilen sonuçlarla belirlenir.Proje, modelin performansını optimize etmek için çeşitli parametre ayarlarının nasıl yapıldığını inceler.Bu ayarlamaların,modelin pistteki performansını artırıp artırmadığını deęerlendirilir.Konvolüsyonel sinir aęlarının mekansal özellikleri, zaman serisi özelliklerinin ise rekürrent sinir aęlarıyla elde edildięi modelin başarısını inceler.Bu yaklaşımın, modelin performansını artırmada etkili olup olmadığı deęerlendirilir.Proje, gerçek dünya verileriyle simülatör verilerinin birleştirilmesinin modelin eğitime nasıl bir etki yaptığını araştırır.Bu yöntemin, modelin gerçek dünyadaki genelleme yeteneğini artırıp artırmadığını deęerlendirilir.Bu sonuçlar, projenin modelin performansını ve gerçek dünyada uygulanabilirliğini deęerlendirmesine olanak tanır.Ayrıca, derin öğrenme ve otonom araç teknolojilerinin gelecekteki gelişimine katkı sağlayacak bilgiler sunar.



Resim 5: Otomatik modda sürüş

Kaynakça

- [1] E. Yiğit, A. E. ONER, and O. Yöntem, “Otonom araçların otomotiv sektörüne etkileri ve beraberinde getirdiği yenilikler,” *Avrupa Bilim ve Teknoloji Dergisi*, pp. 181–186, 2020.
- [2] Z. Chai, T. Nie, and J. Becker, *Autonomous driving changes the future*. Springer, 2021.
- [3] A. Madan and D. Jharwal, “Road lane line detection using opencv,” *International research journal of modernization in Engineering Technology and Sciences (IRJETS)*, 2022.
- [4] Y. Saadna, A. Behloul, and S. Mezzoudj, “Speed limit sign detection and recognition system using svm and mnist datasets,” *Neural Computing and Applications*, vol. 31, no. 9, pp. 5005–5015, 2019.
- [5] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, “A survey on theories and applications for self-driving cars based on deep learning methods,” *Applied Sciences*, vol. 10, no. 8, p. 2749, 2020.
- [6] R. Simhambhatla, K. Okiah, S. Kuchkula, and R. Slater, “Self-driving cars: Evaluation of deep learning techniques for object detection in different driving conditions,” *SMU Data Science Review*, vol. 2, no. 1, p. 23, 2019.
- [7] M. Smolyakov, A. Frolov, V. Volkov, and I. Stelmashchuk, “Self-driving car steering angle prediction based on deep neural network an example of carnd udacity simulator,” in *2018 IEEE 12th international conference on application of information and communication technologies (AICT)*, pp. 1–5, IEEE, 2018.