İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrikli Araç Şarj İstasyonlarına Entegre Plaka Tanıma Sistemi ile Otomatik Şarj Başlatma

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Deniz Uysal

ORCID 0000-0002-5179-9284

Proje Danışmanı: Dr. Öğr. Üyesi Serpil Yılmaz

Şubat 2024

Elektrikli Araç Şarj İstasyonlarına Entegre Plaka Tanıma Sistemi ile Otomatik Şarj Başlatma

Öz

Bu çalışma, elektrikli araç şarj istasyonlarındaki üyelik sistemleri ve OCPP (Open Charge Point Protocol) protokolü üzerine bir araştırma sunmaktadır. RFID kart kullanımının yerine geçebilecek bir alternatif olarak, plaka tanıma sistemi ve OCPP protokolü ile entegrasyonunun teknik açıdan değerlendirilmesini amaçlamaktadır. YOLOv7 modeli kullanılarak geliştirilen plaka tanıma yazılımı, Raspberry Pi üzerinde geliştirilmiştir. Bu yaklaşım, kullanıcıların araçlarını şarj etmek için plakalarını kullanabilmelerini sağlamaktadır. Yapay zekâ algoritmalarının sunduğu hızlı ve etkili plaka tanıma yetenekleri, şarj işlemlerinin otomatikleştirilmesinde teknik bir iyileştirmeyi ifade etmektedir. Öte yandan, OCPP protokolüne uygun bir röleli devre ve buna bağlı bir sunucu yazılımı geliştirilmiştir. Bu entegrasyon, yapay zekâ destekli plaka tanıma verilerinin güvenli bir şekilde yönetilmesini sağlayarak, OCPP standardına uygun şarj işlemlerini simüle etmektedir. Bu çalışma, geleneksel şarj sistemlerine teknik bir alternatif sunma amacını taşımaktadır. Geliştirilen sistem ile, kullanıcıların araçlarını şarj etmek için herhangi bir fiziksel kart, kredi kartı ya da QR kod benzeri bilinen yöntemleri kullanmadan, sadece plakalarını tanıtarak işlemi başlatmalarına olanak tanımaktadır. Yapılan geliştirmeler, elektrikli araç şarj deneyimini teknik ve güvenli bir perspektiften ele alarak iyileştirmeyi amaçlamaktadır.

Anahtar Sözcükler: Elektrikli Araç Şarj İstasyonları, OCPP 1.6 Protokolü, Plaka Tanıma Sistemi, YOLO V7 Modeli, Şarj İstasyonu Ödeme Yöntemleri

Auto Charge with Integrated License Plate Recognition System at Electric Vehicle Charging Stations

Abstract

This paper presents a study on the membership systems in the charging stations for electric vehicles and on the OCPP (Open Charge Point Protocol) protocol. It aims to technically evaluate the licence plate recognition system and its integration with the OCPP protocol as an alternative to the use of RFID cards. The licence plate recognition software was developed using the YOLOv7 model. It was developed on a Raspberry Pi. This approach allows users to use their licence plates to charge their vehicles. A technical improvement in the automation of the charging process is the fast and efficient recognition of licence plates by means of artificial intelligence algorithms. On the other hand, a relay circuit compliant with the OCPP protocol and associated server software have been developed. This integration simulates charging operations according to the OCPP standard and allows the secure management of AI-based number plate recognition data. This study aims to provide a technical alternative to traditional charging systems. The developed system allows users to initiate the process of charging their vehicles by simply identifying their licence plate, without using known methods such as physical cards, credit cards or QR codes. The developments aim to improve the electric vehicle charging experience from a technical and security perspective.

Keywords: Electric Vehicle Charging Stations, OCPP, License Plate Recognition System, YOLO V7 Model, Charging Station Payment Methods

Teşekkür

Proje çalışmasına destek ve katkılarından dolayı danışman hocam Dr. Öğr. Üyesi Serpil Yılmaz' a, konu hakkında değerli katkıları için TTTech-Auto Yazılım AŞ.' den takım liderim Gökhan Bilekdemir ile takım arkadaşlarıma, konu hakkında çok değerli bilgi aktarımları ve destekleri için Vestel Arge ekibinde Akın Acar ve ekibine teşekkürü borç bilirim.

Aynı zamanda, her zaman yanımda olan ve sonsuz destek veren eşim Ayşe Güçlü Uysal' a teşekkür ederim.

İçindekiler

Şekiller Listesi

Tablolar Listesi

Kısaltmalar Listesi

YOLO	You Only Look Once
OCPP	Open Charge Point Protocol
CHAdeMO	Charge de Move
CCS	Combined Charging System
IEC	International Electrotechnical Commission
SAE	Society of Automobile Engineers
SOAP	Simple Object Access Protocol
QR Code	Quick Response Code
RFID	Radio Frequency Identification
EV	Electric Vehicle
CNN	Convulotional Neural Network
COCO	Common Objects in Context
ELAN	Efficient Layer Aggregation Network
E-ELAN	Extended Efficient Layer Aggregation Network
VGG	Visual Geometry Group
ResNet	Residual Network
CTC	Connectionist Temporal Classification
LSTM	Long Short Term Memory

Bölüm 1

Giriş

Elektrikli araçlar, batarya teknolojisi, araç yazılımı ve bağlanabilirlik alanındaki hızlı ilerlemelerle birlikte, içten yanmalı motorlu araçları geçerek daha güvenli, ekonomik ve sürdürülebilir bir ulaşım deneyimi sunma potansiyeli taşımaktadır. Bu gelişmeler, fosil yakıtların neden olduğu zararlı emisyonları azaltma ve çevresel duyarlılığımızı artırma amacını taşımaktadır.

Ancak, elektrikli araçların yaygın olarak benimsenmesini engelleyen bazı endişeler bulunmaktadır. Bu endişelerin başında, elektrikli araçların sınırlı menzili nedeniyle duyulan "menzil kaygısı" gelmektedir. Ortalama bir elektrikli aracın menzili şu anda 350-400 km civarındadır ve bu durum, tüketicilerin elektrikli araçlarıyla uzun mesafe seyahat etme konusundaki çekincelerini artırmaktadır.

Elektrikli araçların daha yaygın olarak kullanılabilmesi için, şarj altyapısının geliştirilmesi önemli bir faktördür. Elektrikli araç şarj istasyonlarının çeşitli seviyeleri bulunmaktadır. Seviye 1 istasyonlar ev prizleri üzerinden şarj imkânı sunarken, Seviye 2 istasyonlar daha yüksek güçlü prizler kullanarak araçları 6-12 saat içinde tam şarj edebilmektedir. Seviye 3 istasyonlar ise DC hızlı şarj imkânı sağlayarak araçları 30 dakikada tam şarj edebilme özelliğine sahiptir, ancak maliyetleri daha yüksektir.

Elektrikli araç şarj altyapısının benimsenmesi için dünya genelinde farklı politika girişimleri bulunmaktadır. Kuzey Amerika'da, Ulusal Elektrikli Araç Altyapısı (NEVI) programı ve Enflasyon Azaltma Yasası gibi politikalar, şarj altyapısının genişletilmesini teşvik etmektedir. Avrupa'da ise Alternatif Yakıtlar Altyapısı Direktifi çerçevesinde, ülkeler kendi programlarını uygulayarak elektrikli araç şarj altyapısını desteklemektedir. Çin ise güçlü liderlik sergileyerek DCFC istasyonlarının kurulu tabanını diğer ülkelerinkinden öne çıkarmıştır.

Elektrikli araç şarj altyapısının geleceği için yapay zekâ (AI) ve makine öğrenimi (ML) gibi teknolojilerin kullanımı da önemlidir. AI, elektrikli araç şarj istasyonlarının güvenliğini artırabilir, otomatik yük dengeleme sağlayabilir ve kişiselleştirilmiş reklamlar gibi katma değerli uygulamalara olanak tanıyabilir. Ayrıca, akıllı otopark ve yol altyapısı gibi akıllı şehir ekosistemlerinin geliştirilmesine de katkı sağlayabilir.

Elektrikli araç şarj altyapısının gelişimi, menzil endişelerini giderme ve sürdürülebilir bir ulaşım geleceği için önemli bir faktördür. Politika girişimleri ve teknolojik yeniliklerle birlikte, elektrikli araçların daha geniş kitleler tarafından benimsenmesi ve sürdürülebilir bir ulaşım paradigmalarının oluşturulması mümkün olacaktır.

Bu çalışma, özellikle üyelik sistemleri, ödeme yöntemleri, plaka tanıma teknolojisinin entegrasyonu ve OCPP, elektrikli araç şarj istasyonlarının teknik ve kullanıcı odaklı olarak gelişimini amaçlamaktadır. Ayrıca, bu alandaki teknolojik gelişmeleri ve gelecekteki perspektifleri anlamak için bir çerçeve sunmaktadır.

Şarj noktası operatörleri, şarj istasyonlarının etkili bir şekilde yönetilmesi ve kullanıcıların enerji ihtiyaçlarına erişim sağlaması bakımından kritik bir rol oynamaktadır. Bu bağlamda, bu çalışma özellikle plaka tanıma teknolojisinin yapay zekâ ile geliştirildiği ve YOLOv7 algoritması kullanılarak oluşturulduğu bir sistem üzerine odaklanarak, kullanıcıların şarj hizmetlerine daha hızlı, güvenli ve standartlara uygun bir erişim sağlamayı amaçlamaktadır.

Plaka tanıma sisteminin YOLOv7 algoritması ile güçlendirilmesi, yapay zekâ destekli hızlı ve etkili plaka tanıma yeteneklerini beraberinde getirmektedir. Bu yaklaşım, şarj işlemlerini otomatikleştirmekte ve kullanıcı deneyimini önemli ölçüde iyileştirmektedir. Sistem, araç plakalarını anlık olarak tanıyarak, kullanıcıların şarj işlemlerini fiziksel kart veya benzeri geleneksel yöntemlere ihtiyaç duymadan başlatmalarına olanak tanımaktadır.

Teknolojik ilerlemeler ve gelecek perspektifi bağlamında, bu çalışma aynı zamanda yapay zekâ destekli plaka tanıma sistemlerinin sağladığı avantajları da değerlendirmeyi amaçlamaktadır. Bu avantajlar arasında hızlı tanıma, düşük hata oranları ve sistem performansındaki sürekli iyileştirmeler, kullanıcıya özel deneyim

gibi faktörler, elektrikli araç şarj istasyonlarının operasyonel verimliliği ve kullanıcı memnuniyeti açısından önemli avantajlar sunmaktadır. Bu çalışmanın, geliştirilen yapay zekâ destekli plaka tanıma sistemleri ile şarj işlemlerindeki etkinliği ve güvenliği artırmaya yönelik katkı sağlaması beklenmektedir.

Bölüm 2

Literatür Araştırması ve Teori

Bu bölümde, çalışma bileşenlerini oluşturan elektrikli araç şarj istasyonları, şarj istasyonlarının yönetimi ve ödeme sistemleri, çalışmanın plaka tanıma sistemi ve teorik altyapısı başlıklar halinde aktarılacaktır.

2.1 Elektrikli Araç Şarj İstasyonları ve Türleri

Elektrikli araçlar da diğer birçok teknoloji gibi laboratuvar ortamında geliştirilmeye başlanmış ve sanayileşmenin de yardımıyla günlük hayatımızdaki yerini almıştır. Elektrikli araç teknolojisinin hayata geçirilmesinde çeşitli donanım ve yazılımlar bir arada kullanılmaktadır. Batarya grubu, elektrik motoru, invertör gibi bu donanımlar araç üzerinde veya dışında yer alabilmektedir. Bunun en güzel örneği aracın bataryalarını şarj etmek için kullanılan şarj cihazıdır. Elektrikli araçların hareket kabiliyetini sürdürebilmesi için bataryaların şarj edilmesi gerekir. Şarj cihazı, bataryalarda enerji depolamak için şebekeden aldığı voltajı gerekli forma dönüştürerek şarjı sağlar (Zheng, Z. Vd., 2011). Bu şarj işlemini yapmak için adımlar şu şekildedir (Gao, Y. vd., 2019): 1. Şarjın başlaması 2. Şarjın tamamlanmasına yakın dengeleme Şarj tamamlanmak üzereyken dengeleme 3. Şarjı durdurma Şarjı durdurma Ayrıca aküler şarj edilirken üç farklı yöntem kullanılabilir. Bunlardan ilki tel ile, ikincisi kablosuz yani herhangi bir temas olmadan elektromanyetik iletim ile ve üçüncüsü ise bataryanın değiştirilmesi ile yapılabilmektedir (He, J. vd., 2018). Literatürde şarj istasyonuna bağlanan batarya grubunun şarj işleminin üç farklı yöntemle gerçekleştirildiği belirtilmektedir. Bunlar sabit frekans yöntemi, şebekeden alternatif akım yöntemi ve bir doğrultucu üzerinden doğru akım ile şarj yöntemidir. Bu makalede kablolu ve kablosuz şarj yöntemleri detaylı olarak ele alınmıştır (Bräunl, T. 2012; Pappas, J. C. K, vd., 2021). 2. Kablolu Şarj Sistemleri Kablolu şarj, bir jeneratörden sağlanan elektriğin standart bir konektörle sonlandırılan iletken bir kablo ile araç üzerindeki şarj konektörüne doğrudan iletilmesini sağlayan bir şarj yöntemidir. Kablolu şarj, tamamen elektrikli veya plug-in hibrit elektrikli araçlarda kullanılır.

Temelde iki farklı şekilde uygulanır: birincisi şarj istasyonlarında (evlerde veya halka açık yerlerde) şarj etmek, ikincisi ise bataryayı tam şarjlı bir batarya ile değiştirmek. Şarj istasyonunda şarj türleri ise AC şarj ve DC hızlı şarj olarak ikiye ayrılmaktadır. (Ekici, Y. E. Vd., 2021)

2.1.1 AC normal şarj

Kablolu şarj yönteminde, şarj işlemi için bir iletken ve bir konektöre ihtiyaç vardır. Kablo, gerekli enerjiyi standart bir prizden veya halka açık bir şarj istasyonundan elektrikli araca aktarır. Bu yöntemin en büyük dezavantajı, kullanıcının kabloyu ve prizi fişe takmasıdır. Kablodaki herhangi bir arıza veya temassızlık istenmeyen sonuçlara yol açabilir. (Dericioğlu, Ç. vd., 2019) Japonya'da CHAdeMO şarj standardı kullanılmaktadır. Dünyanın geri kalanında ise AC şarj standardı olarak SAE belirlenmiştir (Bohn, T., 2019; SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler, 2010). Ancak DC şarj sistemi için henüz resmi bir standart bulunmamaktadır. Ülkeler kendi DC şarj standartlarını kullanmaktadır. Kablolu şarjın belirli prosedürlere göre yapılması gerekmektedir. Bunun için standart olarak üç farklı şarj seviyesi ile şarj işlemi gerçekleştirilmektedir. Aslında şarj seviyelerinin temelde güç seviyelerini tarif ettiği söylenebilir.

2.1.1.1 Seviye 1 Şarj Sistemi

Seviye 1 şarj, araçtaki şarj soketi ve şarj kablosu ile yapılan şarjdır. Gerekli güç, yaygın olarak topraklanmış bir prizden (NEMA 5-15R) elde edilir.

2.1.1.2 Seviye 2 Şarj Sistemi

Seviye 2 şarj genellikle özel veya kamuya açık şarj istasyonlarında kullanılır. Şarj konnektörü, kablolar ve AC şarj altyapısı kullanılır. Bu şarj yöntemi 7 kW (32 Amper tek faz) veya 21 kW (üç faz) "Hızlı AC şarj" olarak adlandırılır. Seviye 2 şarj yöntemi işyerleri veya halka açık şarj istasyonları için de geçerlidir.

2.1.1.3 Seviye 3 Şarj Sistemi

Bu şarj yönteminde, aküler kamuya açık veya özel şarj istasyonlarında doğru akımla şarj edilir. Şarj işlemi gerçekleşirken uygun bir iletken ve uygun bir konektör bulunmalıdır. DC şarj, AC Seviye 1 ve Seviye 2 şarjdan çok daha hızlıdır. Seviye 3 şarj sistemi de hızlı DC şarj olarak düşünülebilir ve araca bağlı olarak 50 kW güç rahatlıkla araca yönlendirilebilir. (Karlsson, P. ve Svensson, J., 2003) Seviye 3 şarj cihazı ile batarya sadece yaklaşık 20 dakika içinde %0'dan %80'e kadar şarj edilebilmektedir. (SAE., 2017)

2.1.2. DC Hızlı Şarj

2012 yılında dünya çapında yaklaşık 2400 DC şarj istasyonu satılmıştır. Bu sayı 2019 yılında 100.000'e yükselmiştir. Pike Research'e göre bu rakamın 2020 yılında 460.000'e ulaşacağı tahmin edilmektedir. (McPhail, D., 2014) DC hızlı şarj cihazları kullanıcılara daha hızlı şarj imkânı sağlarken, şarj istasyonunun yüksek güç talebi nedeniyle elektrik şebekesine ani yük bindirme ve dolayısıyla önemli maliyetlere yol açma potansiyeline sahiptir. Bu nedenle DC hızlı şarj istasyonları bakım, onarım ve servis için yıllık birkaç bin dolar daha pahalıya mal olabilmektedir. (America EN, Phoenix A., 2012). Bu nedenle, şarj istasyonlarının kamuya açık alanlarda optimum konumlarda kurulması, olası yıllık ekstra maliyetlerin azaltılmasına yardımcı olabilir (McPhail, D., 2014). Elektrikli otomobillerin yaygınlaşmasının en önemli nedenlerinden biri de şarj altyapısının yaygınlaştırılması ve geliştirilmesidir. Bir şarj istasyonunun gücü genellikle altyapı yatırımı ile şarj süresinin uzunluğu arasında bir optimizasyon hesaplamasıdır. Şarj altyapısı, AC ve DC şarj olmak üzere ikiye ayrılmaktadır. (Ahmad, A. vd., 2017; Harighi, T. vd., 2018). Ayrıca dünyada elektrikli otomobil şarj sürecini standartlaştırmak ve yasal altyapı sağlamak için çalışan IEC, CHAdeMO ve SAE olmak üzere üç ana kuruluş bulunmaktadır. Tesla kendi tescilli şarj standardını kullanmaktadır.

Elektrikli araçların şarj süresi üç ana faktöre bağlıdır. (Mwasilu, F. vd., 2014) Batarya paketinin boyutu, şarj cihazının güç derecesi ve şarj cihazına o anda bağlı olan elektrikli araç sayısı. (Liu, L. vd., 2015). Seviye 1 ve Seviye 2 şarj işlemleri genellikle

daha uzun sürmekte ve batarya paketlerinin boyutu ve elektrikli araç sayısı arttıkça bu şarj istasyonları yetersiz kalmaktadır. (Channegowda, J. vd., 2015)

Bu nedenle, DC hızlı şarj cihazı bu gereksinimi karşılamak için en umut verici aday olarak öne çıkmaktadır. DC hızlı şarj, elektrikli araç kullanıcıları için olduğu kadar ticari şarj ağları için de gereklidir. DC hızlı şarj IEC CHAdeMO, SAE J1772 Combo ve Tesla Supercharger tarafından sunulmaktadır.

Günümüzde elektrikli araç güç yönetim sistemi ile şarj cihazı kontrol üniteleri arasında güvenli iletişim kurmak için kontrolör alan ağı (CAN) ve PLC protokolleri kullanılmaktadır.

2.2 Şarj İstasyonları Yönetim ve Ödeme Sistemleri

2.2.1 OCPP Protokolü

OCPP protokolü için en kısa tanımlama bir operatör ile şarj cihazı arasındaki iletişimi sağlayan protokoldür şeklinde yapılabilir.

Gelişmekte olan elektrikli araç şarj pazarındaki ürün çeşitliliği, rekabeti artırarak maliyetleri düşürmekte ve mekanik geliştirmeleri teşvik etmektedir. Devlet veya özel şarj tedarikçileri, birçok şarj istasyonu üreticisi ve sistem çerçeve satıcısından seçim yapmaktadır. Bu geniş seçenek, her EV şarj istasyonunun üreticiye veya IT arka uç satıcısına bakılmaksızın merkezi bir sistemle iletişim kurma yeteneğini gündeme getirmektedir. Bu noktada, Açık Şarj Noktası Protokolü (OCPP) devreye girer.

OCPP' nin amacı hem elektrikli araç sürücüleri hem de sistem yöneticileri için çok yönlü ve kullanımı kolay bir çerçeve oluşturarak, gerçek bir şekilde birlikte çalışabilir bir EV şarj altyapısı sağlamaktır.

OCPP ile kullanıcılar, birkaç farklı tedarikçinin şarj istasyonlarını aynı IT arka uç çerçevesine entegre edebilirler. Aynı zamanda en uygun şarj istasyonu tedarikçilerini ve en uygun IT arka uç sağlayıcısını özgürce seçebilirler. OCPP, Open Charge Alliance tarafından geliştirilmiş ve 50'den fazla ülkede ve 10.000'den fazla şarj istasyonunda kabul görmüş bir protokol ve gerçek bir standart haline gelmiştir.

Sınırlayıcı iletişim protokollerinin aksine, OCPP açık ve herhangi bir lisans ücreti veya şartla bağlantılı değildir, bu da benimsemeyi kolaylaştırır.

Birçok şarj noktası satıcısı ve IT çerçeve sağlayıcısı tarafından benimsenen OCPP uyumluluğu, yatırımcılar arasında "olmazsa olmaz" bir gereklilik haline gelmektedir. OCPP' yi benimsemek, altyapı sağlayıcısını tarafsız tutmak ve maliyetleri düşürmek için bir strateji olarak görülmektedir. Open Charge Alliance, OSCP (Open Smart Charging Protocol) adını verdiği başka bir protokolü de piyasaya sürdü. Bu protokol, Şarj Noktası Operatörüne yerel kapasitenin 24 saatlik bir tahminini iletmek için kullanılabilir. Hizmet Sağlayıcı, şarj profillerini erişilebilir sınırlar içinde sığdırmak için elektrikli araçların şarjını planlayacaktır. Bu, şarj noktası yönetim sistemi ile saha sahibinin enerji yönetim sistemi arasında bir protokoldür. (Pruthvi, T. V. vd., 2019)

2.2.1.1 OCPP Sürüm Geçmişi

OCPP'nin ilk sürümünden bu yana temel olarak üç farklı sürümü bulunmaktadır. Bunlar sırasıyla OCPP 1.5, OCPP 1.6 ve OCPP 2.0'dır.

2.2.1.1.1 OCPP 1.2

OCPP'nin bu sürümü daha az işlevsellikle OCPP 1.5'e benzer.

2.2.1.1.2 OCPP 1.5

OCPP 1.5'te (OCPP 1.5, 2012) tanımlanan 25 İşlem vardır, 10'u Şarj İstasyonu tarafından başlatılır ve 15'i Merkezi yönetim sistemi tarafından başlatılır. Yetkilendir, Önyükleme Bildirimi, Veri Aktarımı, Teşhis Durum Bildirimi, Ürün Yazılımı Durum Bildirimi, Kalp Atışı, Sayaç Değerleri, İşlemi Başlat, Durum Bildirimi, İşlemi Durdur, şarj noktası tarafından başlatılır ve Rezervasyonu İptal Et, Kullanılabilirliği Değiştir, Yapılandırmayı Değiştir, Önbelleği Temizle, Veri Aktarımı, Yapılandırma Al, Teşhis Al, Yerel Liste Sürümünü Al, Uzaktan Başlatma İşlemi, Uzaktan Durdurma İşlemi, Şimdi Rezerve Et, Sıfırla, Yerel Liste Gönder, Bağlayıcı Kilidini Aç, Ürün Yazılımını Güncelle, merkezi sistem tarafından başlatılır.

2.2.1.1.3 OCPP 1.6

OCPP' nin bu ayarlaması, internet üzerinden bölümler arasında mesaj göndermek için SOAP Çerçevesini kullanır. SOAP' ın avantajı, mesaj gönderme ve tolere etme iş yerlerinin standart (OCPP 1.6, 2017) tarafından sabitlenmiş olmasıdır. Bir SOAP mesajının içeriği Genişletilebilir Biçimlendirme Dili (XML) standardına göre çekilir. Bu lehçe HTML ile özdeşleştirilmiştir. Yazılı içeriğin yanı sıra, XML mesajları aynı şekilde resim ve çalıştırılabilir kod içerebilir. En büyük avantajı mesajların deşifre edilebilir içerikte gönderilmesidir. (You, X., ve Zhao, C., 2017)

2.2.1.1.3 OCPP 2.0

OCPP 2.0, Nisan 2018'de yayınlanan en son uyarlamadır ve 116 kullanım senaryosunda değerlendirilen çok sayıda yeni özellik içerir. OCPP 2.0 sadece JSON' u desteklemektedir. Cihaz Yönetimi, Geliştirilmiş İşlem bakımı, Eklenen Güvenlik, Eklenen Akıllı Şarj işlevleri, 15118 desteği, Görüntüleme ve bilgilendirme desteği ve elektrikli araç şarj ağı tarafından talep edilen çok sayıda ekstra yükseltme gibi birçok dahil edilmiş ve geliştirilmiş işlev vardır. (OCPP 2.0.1, 2020)

2.2.1.2 OCPP Protokolünün Faydaları ve Etkileri

Şarj istasyonu üreticileri, şarj noktası operatörleri gibi çok sayıda aktörün olduğu senaryoda, bir birlikte çalışabilirliğin sağlanması ya tescilli bir standarda ya da açık bir standarda dayanır. Bu iki standart türü arasındaki temel fark, açık standart hareketinin, birlikte çalışabilirliğin işleyişinde herhangi bir tekelci gücün ortaya çıkmasını önleyerek paydaşları arasında endüstri çapında veya ülke çapında birlikte çalışabilirliği tesis etmeyi amaçlamasıdır.

OCPP' nin başlıca avantajlarından bazıları şunlardır:

- Bir şarj noktası sahibinin istediği zaman şebeke operatörünü değiştirmesini desteklemesi

- Şarj istasyonu ve şebeke hizmet sağlayıcısı arasındaki ortak iletişimin, şebeke hizmetlerinin (örn. talep yanıtı) maliyet etkin bir şekilde sağlanması için de kullanılmasına izin verilmesi.

- Şarj istasyonlarına, dolaşım ve faturalandırma hizmetlerine tek tip erişim sağlayarak müşterileri elektrikli araç sahibi olmaya teşvik etmek.

2.2.2 Mevcut Şarj İstasyonları Ödeme Yöntemleri ve Değerlendirmeler

EV şarjı için ödeme yapmanın birkaç yolu vardır. Bunlar, QR kodları, mobil uygulamalar, RFID Kartlar ve Etiketler ve son olarak kredi kartı ile temassız ödemelerdir. Aşağıda başlıklar halinde mevcut sistemler incelenmiştir. (Au, M. vd., 2015; Reznichenko, M., 2023)

2.2.2.1 EV Sürücüleri İçin Mobil Uygulama ile Başlatma

EV şarjı için ödeme yöntemi olarak bir uygulama kullanmanın birkaç avantajı vardır:

Güvenlik: Şarj uygulama sağlayıcıları için güvenlik en üst önceliktir. Şifreleme ve iki faktörlü kimlik doğrulama dahil olmak üzere birçok güvenlik önlemi, ödeme ve kişisel verilerinizi korumak için uygulanır. Bu, EV şarjı için ödeme yapmak için bir uygulama kullanmanın güvenli bir seçenek olmasını sağlar.

Bakiye Kontrolü: Bir uygulama, harcamalarınızı takip etmenizi, kullanımınızı ve şarj geçmişinizi izlemenizi sağlar. Bu, şarj harcamalarınızı daha etkili bir şekilde planlamanıza ve bütçe yapmanıza yardımcı olabilir.

Hızlı ve Kolay Kullanım: Telefon üzerinden bakiye yüklemek son derece hızlıdır ve sadece bir tıklama ile kullanılabilir. Bu, aceleyle veya hareket halindeyken bakiye yüklemeniz ve aracınızı şarj etmeniz için kullanışlıdır.

Konfor: Şarj uygulaması kullanmak pratiktir. Ayrı bir ödeme kartını taşımak veya uygun değişimi bulma konusunda endişelenmenize gerek yoktur. Bunun yerine, sadece akıllı telefonunuzu kullanarak ödeme yapabilir ve aracınızı şarj edebilirsiniz.

Ayrıca, birçok şarj uygulaması, konum tabanlı şarj istasyonu bulucuları sunarak nerede olursanız olun bir şarj istasyonu bulmanıza olanak tanır.

2.2.2.2 RFID Kartları ve Etiketler ile Şarj Başlatma

Eğer genel şarj için telefonunuzu kullanmak istemiyorsanız, şarj noktası işletmecisi tarafından sağlanan fiziksel bir RFID kartı veya anahtarlık kullanılabilir. Bu kart, kimlik bilgilerinizi içerir ve şarj istasyonundaki okuyucuda sürükleyerek kolayca ödeme yapmanıza olanak tanır. Bir şarj uygulaması gibi, şarj ihtiyaçlarınıza bağlı olarak üyelik veya anlık ödeme seçenekleri arasında seçim yapabilirsiniz. Ancak, tüm şarj istasyonlarının aynı RFID kartı ile uyumlu olmadığını unutmamak önemlidir, bu nedenle çeşitli ağlardan şarj istasyonları kullanıyorsanız birden çok kartınız olabilir. Ayrıca, çoğu modern şarj cihazının bir uygulama aracılığıyla kilidini açabilse de bazı modellerin erişim için RFID kartı veya anahtarlık gerektirebileceğini unutmamak önemlidir. Bu nedenle, genel bir şarj istasyonu ararken bir uygulama yeterli olup olmadığını kontrol etmek önemlidir.

2.2.2.3 QR Kodları: Tarama ve Şarj Başlatma

QR kodları, EV şarj istasyonlarına erişimi sağlamak için popüler bir yöntemdir. Cihaz üzerindeki ekranlara yerleştirilen QR kodun şarj noktası operatörüne ait ya da kuruluşun sunucularına bağlanabilen bir uygulama yardımıyla okutulması prensibiyle çalışmaktadır. Bu yöntem hem kullanıcılar hem de sağlayıcılar için çeşitli avantajlar ve dezavantajlar sunmaktadır. Bu avantajlara ve dezavantajlar ise aşağıdaki gibi özetlenebilir.

Konfor: QR kodları, sadece bir akıllı telefon ve internet erişimi ile EV sürücülerinin şarj istasyonlarını bulmasını ve kullanmasını kolaylaştırır, özel bir erişim kartı veya anahtar gerektirmez.

Erişilebilirlik: Ancak aynı zamanda, QR kodlarının kullanıcıların bir akıllı telefona ve internet erişimine sahip olmalarını gerektirdiği unutulmamalıdır. Bu durum bir uygulama için de geçerlidir. Bu, sınırlı internet bağlantısı veya akıllı telefon sahipliği olan bölgelerde şarj istasyonlarının erişilebilirliğini sınırlayabilir.

Maliyet Avantajı: QR kodları, şarj istasyonu sağlayıcıları için erişimi yönetmek için pahalı donanım veya yazılım gerektirmez, bu nedenle maliyet etkin bir yol sunar.

Güvenlik Sorunları: Ancak, deneyim gösteriyor ki, bu tür bir yöntem, siber saldırı ve diğer güvenlik ihlallerine karşı savunmasız olabilir, bu da şarj ağının güvenliğini ve güvenilirliğini tehlikeye atabilir.

Özetle, QR kodları, EV şarjı için bir dizi avantaj sunsada bir dizi zorluk ve kısıtlamayla birlikte gelir. Şarj istasyonu sağlayıcıları, kullanıcılar için en iyi deneyimi sağlamak için bu faktörleri dikkatlice düşünmelidir.

2.2.2.4 Kredi Kartı ile Temassız Ödeme

Bu ödeme yöntemi, ağa özgü bir abonelik veya bir uygulama indirmeyi gerektirmez. Bunun yerine, EV sürücüleri, her zaman yanlarında taşıdıkları standart kredi kartları ile şarj oturumlarını ödeyebilirler. Çoğu büyük şarj noktası sağlayıcısı, EV sürücülerinin ödeme yapmalarını kolaylaştırmak için temassız kredi veya banka kartı ödeme seçenekleri sunar. Ancak, bu ödeme yönteminin müşteriler için genellikle en pahalı olanı olduğu, çünkü şarj noktası sahipleri için maliyetli bir kurulum gerektirdiği ve bu masrafları EV sürücüsünün karşıladığı unutulmamalıdır. Ayrıca, operatörler ödeme yöntemlerini daha uygun hale getirmek için çalışsalar da temassız ödeme genellikle yüksek ve ultra-hızlı cihazlarda mevcut olabilir çünkü kurulum maliyetleri yüksektir.

2.3 YOLO Derin Öğrenme Algoritması İncelemesi

Görüntüler kullanarak bilgi elde etmek amacıyla nesne tespiti sistemlerinin geliştirilmesi, akıllı trafik gözetim sistemleri gibi, görüntü işleme ve bilgisayar görüsüne yeniden ilgi uyandırdı. Bilgisayar görüsü alanı, bilgisayarların "bakma" yeteneğine odaklanır. Bu, bilgisayarların ve diğer cihazların nesne tespiti ve ardından görüntü işleme teknikleri için insan gözlerini yerine koymalarına izin verir. Birçok çalışma ve yöntemin yerine konulmasıyla YOLO, trafik yönetimini, plaka tanıma ve nesne tespitini geliştirmek için en iyi Algoritmalar arasında yer almaktadır. YOLO, CNN tabanlı bir tekniktir; CNN, derin sinir ağlarının bir alt kümesidir ve görüntüleri algılamak ve sınıflandırmak için kullanılır. Bu algoritmalar, akıllı şehirlerin trafik

levhalarını, araçları ve insanları tanımlamasına olanak tanır. CNN'nin gerçek değeri, eğitim aşamasını takip eden aşamada, neredeyse hiç insan müdahalesi olmadan kritik unsurları otomatik olarak tanıyabilmesidir. En yüksek doğruluk seviyesini en hızlı işleme hızında elde etmek için birçok farklı CNN tasarımı geliştirilmiştir. En bilinen ve yaygın olarak kullanılan yöntemler arasında Fast R-CNN, Faster R-CNN, Bölge Tabanlı Evrişimli Sinir Ağları (RCNN), Bölge Tabanlı Tam Evrişimli Ağlar (R-FCN), Tek Atış Dedektörü 13 Multibox (SSD), Mekansal Piramit Havuzlama (SPP-net) bulunmaktadır. Bir çalışmaya göre (Girshick, R., 2015), nesne tespiti için Fast R-CNN yöntemine dayanmaktadır. Yeni önerilen model, derin evrişimli ağlar kullanarak nesneleri etkili bir şekilde sınıflandırır. Fast RCNN, eğitim ve test hızını artırmak, algılama doğruluğunu artırmak için bir dizi yenilik kullanır. Ancak Fast R-CNN, bir nesneyi büyük bağlamı göremediği için bir görüntüde arka plan yamalarını nesne olarak hatalı bir şekilde algılar. Aynı zamanda YOLO, Fast R-CNN'ye kıyasla yarıdan daha az arka plan hatası yapar. (Redmon, J. vd., 2015) YOLO, nesne tespitini regresyon veya sınıflandırma içeren bir süreç olarak karakterize etmeye çalışan birçok çalışma ve SSD öncesi, (Szegedy, C. vd., 2013) YOLO' yu basit bir sınırlayıcı kutu çıkarımı kullanarak test görüntüsü için ikili bir maske oluşturan ve tespitleri çıkaran bir Derin Sinir Ağı tabanlı regresyon olarak tanımlamıştır. Öte yandan, model çakışan nesnelerle başa çıkmakta zorlanır ve düz bir biçimde artırılan sınırlayıcı kutular neredeyse kusursuz değildir. YOLO, giriş görüntüsünü kullanarak S × S bir ızgara oluşturur. Bir nesnenin merkezi belirli bir ızgara hücresine denk gelirse, o ızgara hücresi o belirli nesneyi tanımlamaktan sorumludur. Her ızgara hücresinde B, her kutu için sınırlayıcı kutular ve güven skorları tahmin edilir. Bu güven skorları, modelin bir kutunun bir nesne içerdiğine olan güvenini ve kutunun ne kadar doğru tahmin edildiğini gösterir.

Güven, resmi olarak Pr(Nesne) $*$ IOUpred gerçek , olarak tanımlanır ve nesnelerin var olma olasılığını gösterir (Pr(Nesne) ≥ 0) ve tahminin güvenini gösterir (IOUpred gerçek). Aynı zamanda, kutu sayısından bağımsız olarak, her ızgara hücresinde C koşullu sınıf olasılıkları Pr(Sınıf i | Nesne) de tahmin edilmelidir. Sadece bir nesneyi içeren ızgara hücresinden gelen katkı hesaplanır. Test zamanında, her kutu için sınıf özgü güven skorları, bireysel kutu güven tahminlerinin koşullu sınıf olasılıkları ile çarpılmasıyla elde edilir, şu şekildedir: (Ogunboye, O. E. ,2023).

Pr(Object) $*$ IOUpred doğruluk $*$ Pr(Class i | Object) (1)
= Pr(Class i) $*$ IOUpred doğruluk (2)

Burada; Pr(Nesne) kutunun bir nesne içerme olasılığıdır. IoU, tahmin edilen kutu ile gerçek değer arasındaki kesişim-birleşim oranıdır. Pr(Sınıf i | Nesne) ise bir nesnenin var olduğu varsayıldığında nesnenin bir sınıfa ait olma olasılığıdır. Model eğitimi sırasında kayıp fonksiyonları, bu metrikler kullanılarak optimize edilir. YOLO modelinin eğitiminde kullanılan kayıp fonksiyonları genellikle sınıflandırma kaybı (Lsınıflandırma), güven kaybı (Lgüven) ve sınırlayıcı kutu konumlandırma kaybı (LCIoU) içerir. (Redmon, J. vd., 2017)

Hesaplama denklemleri aşağıdaki gibidir:

$$\text{LOSS} = \text{Lsınıflandırma} + \text{Lgüven} + \text{LCIoU} \quad (3)$$

$$Lclassification = \sum_{i=0}^{s^2} \ell_i^{obj} \sum_{j=0}^{B} [\, (p_i(c) - \widehat{p_i}(c))^2 \,] \quad (4)$$

$$\text{Lconfidence} = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \ell_i^{obj} \left[\left(\text{C}_i - \hat{\text{C}}_i \right)^2 \right] + \lambda_{no\ obj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \ell_i^{no\ obj} \left[\left(\text{C}_i - \hat{\text{C}}_i \right)^2 \right] \quad (5)$$

Denklem (4)'te, $\ell i\ obj$, bu ızgarada bir nesne merkezinin olup olmadığını belirleme anlamına gelir. Eğer ızgara bir nesne merkezine sahipse, nesnenin kategori olasılığını tahmin etme görevine sahiptir. Denklem (5)'te, Ci güven skorunu temsil eder, Ci tahmin kutusu ile sınırlayıcı kutu arasındaki kesişimi gösterir ve λnesne dışı sınıflandırma hatasının ağırlığını temsil eder.

A. Sınıflandırma kaybı: Sınıflandırma kaybı, bir nesne tespit edildiğinde sınıf olasılıklarındaki kare hatadır. Yukarıdaki Denklem (4), sınıflandırma kaybını (Agni) gösterir.

B. Konumlandırma kaybı: Konumlandırma kaybı fonksiyonu, Projekte edilen sınırlayıcı kutu konumlarındaki ve boyutlarındaki hataları koordinatlara bağlı olarak ölçer, yani (x, y, w, h). Konumlandırma kaybı fonksiyonu, kaybın önemine göre ağırlık olarak tanımlanan lambda koordinatları varsa sağlanır.

C. Güven kaybı: Sınırlayıcı kutu tahminleri, koordinatları (x, y, w ve h) ve bir güven değeri içerir. İki kayıp fonksiyonu arasında sadece küçük bir fark olduğu beklenebilir. Konumlandırma kaybı, konumlandırma parametrelerine bağlı olduğu için bir güven ölçüsüne sahip olmak önemlidir.

2.3.1 YOLO Eğitimi ve Doğrulama Metrikleri

A. Eğitim Kaybı

Eğitim kaybı metriği, derin öğrenme modelinin eğitim verilerine ne kadar iyi uyduğunu değerlendirir. Başka bir deyişle, modelin eğitim setindeki hata oranını değerlendirir. Eğitim seti, modeli başlangıçta eğitmek için kullanılan veri setinin bir alt kümesidir. Hesaplama açısından, eğitim kaybı, eğitim süreci sırasında yapılan tüm hataların toplanmasıyla belirlenir. Eğitim kaybı, her işlenen yığın sonrasında hesaplanır. Bu, eğitim kaybının grafiksel olarak bir eğri olarak görüntülenmesiyle gösterilir.

B. Doğrulama Kaybı

Bir derin öğrenme modelinin doğrulama setinde iyi performans gösterme yeteneği, doğrulama kaybı metriği kullanılarak ölçülür. Doğrulama seti olarak bilinen veri setinin bir alt kümesi, modelin doğruluğunu doğrulamak için ayrılmıştır. Doğrulama kaybı, eğitim kaybına benzer çünkü her iki metrik de modeldeki hataları belirlemek için doğrulama setindeki her veride yapılan tüm hataları toplamak için kullanılır. Ayrıca, doğrulama kaybı, modelin daha fazla ayarlama veya düzeltme gerekip gerekmediğini veya zaten mükemmel olup olmadığını göstermek için her epok sonrasında değerlendirilir. Genellikle, bu grafiksel olarak bir doğrulama kaybı eğrisi görüntülenerek gösterilir. Doğrulama ve eğitim kaybının kontrol edilme amacı, modelin performansını kesin bir şekilde teşhis etmektir. İhtiyaç duyulması durumunda modeli "İyi Uyumlama" sağlamak için fine-tuning yapılması veya "aşırı uyumlama" veya "yetersiz uyumlama" durumlarını kontrol etmek için yapılmaktadır. Bu parametreler kısaca aşağıda açıklanmıştır;

I. İyi Uyumlanmış Model, hem eğitim kaybının hem de doğrulama kaybının azalmaya ve stabilize olmaya başladığı bir noktaya ulaştığı durumdur.

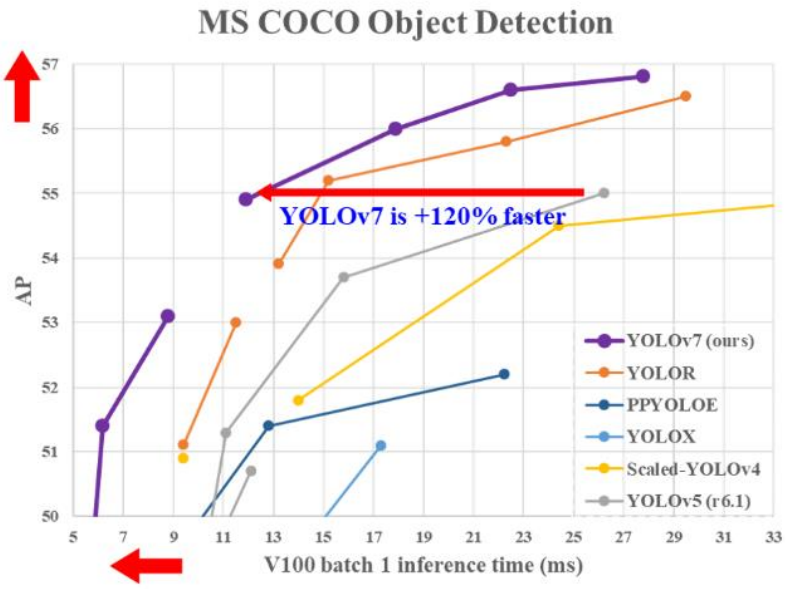
II. Aşırı Uyumlanmış Model, modelin eğitimde kullanılan veriye uygulandığında iyi performans göstermesine rağmen, doğrulama setinde kullanılan yeni veriye uygulandığında kötü performans gösterdiği durumdur. Bu durum, modelin veri için karmaşık veya aşırı uzun süre eğitilmiş olması nedeniyle ortaya çıkar ve buna büyük ölçüde katkıda bulunur. Belirli bir noktadan sonra geçerlilik kaybı düşer ve bir süre

sonra tekrar yükselmeye başlar. Eğitim, kayıp düşük ve stabil olduğunda durdurulabilir; buna "erken durdurma" denir.

III. Yetersiz Uyumlanmış Model, bir modelin eğitildiği veriyi doğru bir şekilde temsil etmekte başarısız olduğu durumdur. Sonuç olarak, önemli bir hata üretir. Doğrulama kaybının eğitim kaybından yüksek olduğu durumların birkaç örneğinde meydana gelir. Bu durum, modelin veriye yeterince uygun olmadığını gösterebilir.

2.3.2 YOLO V7

YOLO'nun en son sürümü olan YOLOv7, hızını ve hassasiyetini artırmak için birkaç mimari değişiklik içermektedir. YOLOv7'nin temel yapıları, ImageNet verileriyle önceden eğitilmemiştir. Bunun yerine, COCO veri kümesi kullanılmıştır. Microsoft'un Common Objects in Context (COCO) Veri Kümesi, 200,000'den fazla görüntü içermekte olup model eğitimi için kullanılmıştır. YOLOv7 Scaled YOLOv4'ü aynı kişilerin yazdığından dolayı benzerlikler beklenmektedir. Ancak, (Wang, C. vd., 2022)'ye göre, YOLOv7, mevcut en hızlı ve en doğru gerçek zamanlı nesne tespit modelidir ve %56.8 doğruluğa sahiptir. YOLOv7, SWINL ve Cascade-Mask R-CNN gibi transformer tabanlı detektörleri her metrikte geride bırakmıştır. Gerçek zamanlı nesne tespiti için YOLOv7 şu anda standart haline gelmiş olup model, ONNX ve TensorRT'ye dönüştürülerek optimizasyon yapılabilir, bu da çıkışı artırır ve kenar cihazlarının işlevselliğini sağlar. Ayrıca, model karmaşıklığı ve hesaplama yoğunluğu iki temel tasarım düşüncesi olarak belirlenmiştir. Ayrıca, giriş/çıkış kanal oranının ve eleman bazlı işlemin ağ çıkarma hızı üzerindeki etkisini incelemek için VovNet (CNN, DenseNet'i son özellik haritasında sadece bir kez birleştirerek daha verimli hale getirmeye çalışır) ve CSPVoVNet modelleri, ELAN'ı genişletmiş ve ona "E-ELAN" adını vermiştir. ELAN'ın başlıca faydası, gradient yolunu düzenleyerek daha derin bir ağın daha hızlı öğrenmesine ve uyum sağlamasına izin vermesidir.



Şekil 2.1: YOLO V7 Eğitim Sonuçları

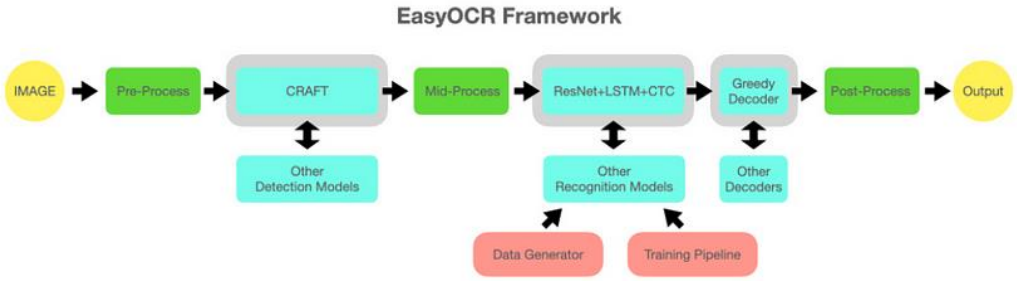
E-ELAN, hesaplama bloğu tasarımını büyük ölçüde değiştirmiş, ancak geçiş katmanı mimarisi hiç değişmemiştir. Orijinal gradient yolunun bütünlüğünü bozmadan, genişleme, karıştırma ve birleştirme yöntemleriyle ağın öğrenme kapasitesini artırır. Tüm hesaplama bloklarına aynı grup parametresi ve kanal çarpanının uygulandığı grup konvolüsyonu, kanal ve hesaplama bloklarının sayısını artırır. Ardından her hesaplama bloğu tarafından üretilen özellik haritaları karıştırılır ve birleştirilir. Sonuç olarak, her özellik haritası grubundaki kanal sayısı, orijinal mimarideki kanal sayısına eşit olacaktır. Son olarak, bu özellik haritası grupları birleştirilir. E-ELAN, daha fazla çeşitlilikte özellik öğrenme yeteneği kazanmıştır. Bu modelin mimari reformu sırasında model ölçekleme yöntemi ve birleştirme tabanlı model göz önüne alındı. Daha sonra çapraz birleştirildi ve bunun nedeni modelin farklı hızlara ve çıkarsamalara ulaşmasıdır. Modelin eğitimden sonra iyileştirilmesinde kullanılan bir yöntem olan "re-parametrization" (yeniden-parametreleme) kullanılmıştır. Daha uzun eğitim süresine rağmen, çıkarsama sonuçları iyileştirilmiştir. Modeli yeniden parametrelemek için iki yöntem kullanılabilir. Model seviyesi ve modül seviyesi toplama yeniden parametrelendirmesi olmak üzere iki tür yeniden parametrelendirme bulunmaktadır. Birincisi, farklı eğitim verileriyle aynı parametrelerle birden fazla benzer model eğitip tüm modellerin ağırlıklarını ortalamaktır. İkincisi ise aynı parametrelerle farklı eğitim verileriyle birden fazla model eğitmektir. Model, farklı epoklarda veri ile eğitilerek modellerin nihai eğitilmiş ağırlıklarını elde etmek için ağırlıkların ortalaması alınır.

Modül seviyesinde yeniden parametrelendirme, model eğitim sürecini aşamalara böler. Çıkışlar, nihai modeli oluşturmak için simüle edilir. YOLOv7'nin yeniden parametrelendirilmiş konvolüsyon mimarisi, kimlik bağlantısı olmadan (RepConvN) RepConv kullanır. Reparametrize edilmiş konvolüsyon, kimlik bağlantılarını ortadan kaldırmak için konvolüsyon katmanlarını artıklar veya birleştirmelerle değiştirir. YOLOv7'nin ana başlığı nihai çıkış başlığıdır. İkincil Başlıklar, orta katman eğitimine yardımcı olur ve Etiket Atayıcı, zemin gerçekliği ve ağ tahmin sonuçları kullanılarak yumuşak etiketleri atar. Bununla birlikte, YOLOv7 ağının Ana Başlığı sonucu tahmin ederken, Yumuşak etiketler nihai sonuçları kullanarak oluşturulur. Temel nokta, aynı yumuşak etiketlerin hem ana hem de ikincil başlıklar için kayıp hesaplamak için kullanılmasıdır.

2.4 EasyOCR Optik Karakter Tanıma Modülü

EasyOCR, Python programlama dilinde kullanılan bir Optik Karakter Tanıma (OCR) modülüdür ve hem esnek hem de kullanımı kolaydır. OCR teknolojisi, veri girişi otomasyonu ve görüntü analizi gibi çeşitli görevler için kullanışlıdır. Bu teknoloji, bilgisayarların fotoğraflardan veya taranmış belgelerden metin tanımlamasına ve çıkarmasına olanak tanır. EasyOCR, OCR veya bilgisayar görüşü konusunda hiçbir geçmişi olmayan geliştiriciler için bile uygulamayı daha kolay hale getirmeye odaklanmasıyla dikkat çeker. Kütüphane, çoklu dil desteği, önceden eğitilmiş metin tespiti ve tanımlama modelleri, hız ve verimlilik odaklı kelime tanıma içinde hız ve verimlilik üzerine odaklanmıştır. (Mahajan, A., 2023)

EasyOCR, yazı tipleri ve metin düzenlerini işleme konusundaki esnekliği, doğruluk ve hızına odaklanması nedeniyle Python geliştiricileri için güvenilir bir seçenektir. Modül, fotoğraflardan metin çıkarma sürecini çeşitli Python projelerinde, masaüstü yazılımlar, çevrimiçi uygulamalar ve diğerleri dahil olmak üzere kullanmak daha basit hale getirir.



Şekil 2.2: EasyOCR Algoritması

EasyOCR' ı Python programlarına entegre ederek metinle ilgili işlemleri kolayca otomatikleştirebilir, taranmış belgelerden veri çıkarmayı iyileştirebilir ve görüntü analizi projelerinizde metin tanıma yeteneklerinden yararlanabilirsiniz. Bu, bilgisayar görüşü uygulamaları için kullanışlı bir araçtır ve Python projelerinizde OCR' ı kullanmanın anlaşılabilir bir yoludur.

Bileşenler EasyOCR, üç bileşenden oluşur: özellik çıkarma, dizi etiketleme ve çözme. Giriş görüntüsünden yararlı özellikleri çıkarmak için özellik çıkarma aşamasında ResNet ve VGG gibi derin öğrenme modelleri kullanılır. Bu özellikler, resimlerde metin tanıma için esastır. Bir sonraki adım olan dizi etiketleme, çıkarılan özelliklerin sıralı bağlamını yorumlamak için Uzun Kısa Vadeli Bellek (LSTM) ağlarını kullanır. LSTM ağları, metin deseni tanıma ve yapılandırma görevleri için önemlidir. Son olarak, çözme kısmı etiketlenmiş dizileri Connectionist Temporal Classification (CTC) algoritması kullanarak gerçek tanınan metne çözer ve transkribe eder. Bu üç öğe, EasyOCR'ın görüntülerden metin çıkarmayı güvenilir ve etkili bir şekilde gerçekleştirmesine olanak tanır. Eğitim boru hattı, metin tanıma üzerine güçlü bir temel sunan deep-text-recognition-benchmark çerçevesine dayanmaktadır.

- Özellik Çıkarma (Resnet ve VGG):

Tanıma modelinin ilk adımı, özellik çıkarma sürecidir. Ek analizler için kullanılabilecek bir özellik kümesi oluşturmak için giriş verileri dönüştürülmelidir. EasyOCR ile bunun için VGG ve Resnet kullanılır.

Resnet, Yedek Ağlar olarak da bilinen, belirli katmanları atlayarak kısayollar veya geçiş bağlantıları kullanarak çalışan bir tür evrişimli sinir ağı (CNN) türüdür. Bu, ağın daha derin olmasını ve hala eğitilebilir olmasını sağlar, çünkü kaybolan gradient sorununu çözer. Resnet mimarisinin temel prensibi, sinyal temsilini doğrudan değil

artık temsil fonksiyonlarını öğrenmektir. Bu, modelin karmaşıklığını azaltır ve ağın öğrenme sürecini kolaylaştırır.

Başka bir CNN türü ise Visual Geometry Group veya VGG olarak adlandırılır. Homojen mimarisi ve basitliği ile bilinir. Homojen mimarisi, birbirinin üstüne konulmuş çok küçük (3x3) evrişim filtrelerinden oluşur. Bu, ağın anlamını ve değişikliğini basitleştirir, çünkü hesap ve parametre sayısını azaltır.

- Dizi Etiketleme (LSTM):

Özellik çıkarmadan sonra sıradaki adım dizi etiketlemedir. Bu işlem için Uzun Kısa Vadeli Bellek (LSTM) ağları kullanılır.

Uzun Kısa Vadeli Bellek (LSTM) ağları, uzun dizgi öğrenme ve bellek türündeki tekrarlayan sinir ağları (RNN'ler) çoklu zaman adımlarının zamansal bağımlılıklarını modelleyebilir. LSTM'nin geleneksel RNN'lere kıyasla farklı tasarımı, kaybolan gradient sorununu önlemeye yardımcı olur. Bellek hücresi ve üç farklı türde kapı - giriş, unutma ve çıkış - bunu başarmasına olanak tanır. Bu öğeler, LSTM'yi uzun diziler boyunca hücre durumundan bilgi ekleyip çıkarmasını sağlayarak dizi etiketleme görevleri için büyük etkinlik sağlar.

- Çözme (CTC):

Çözme, tanıma modelinin son adımıdır ve bunun için Connectionist Temporal Classification (CTC) kullanılır.

CTC adı verilen bir tür kayıp fonksiyonu, zamanın belirsiz olduğu dizi sorunlarına uygulanır. Bu, genellikle konuşma ve el yazısı tanıma gibi durumlarda etiketler ile giriş verisi arasındaki hizalamadan emin olunamadığı durumlarda kullanılır. CTC, mevcut etiketlere ek olarak boş bir etiket ekleyerek değişken uzunluktaki bir tahmin sağlamak için kullanılır. Bu, OCR çözme görevleri için uygundur çünkü girişin hedef dizilerle olası tüm hizalamalarını toplayarak kaybı hesaplar.

EasyOCR'ın tanıma yürütmesi için eğitim işlem hattı (pipeline) olarak değiştirilmiş bir deep-text recognition-benchmark mimarisi kullanılır. Bu çerçeve kullanılarak metin tanıma modelleri çeşitli veri kümeleri üzerinde eğitilebilir, bu da EasyOCR'ı son derece esnek ve etkili kılar.

Bölüm 3

Çalışma Ortamı ve Metodoloji

Bu başlıkta projenin gerçekleştirilmesi, üç ana başlık altında incelenecektir. Bu ana başlıklar şu şekildedir.

- Çalışmanın uygulamasının yapıldığı Raspberry Pi3 ve üzerinde yapılan çalışmalar
- Şarj İstasyonunu simüle eden yazılımın tanıtılması ve geliştirme aşamaları
- Plaka tanıma sistemini oluşturan komponentlerin kullanımı ve bu komponentlerde yapılan çalışmalar.

3.1 Çalışma Ortamı

3.1.1 Raspberry Pi 3B

Raspberry Pi, İngiltere merkezli Raspberry Pi Foundation tarafından geliştirilen bir devre kartı türüdür. Eğitsel kullanım için yapılmış bir tek kartlı bilgisayardır.

Raspberry kendine özgü bir işletim sistemi desteği de sunmaktadır. Sunduğu işletim sistemi Linux tabanlı bir sistem olduğundan çalışmamızda bize esneklik ve kolaylık sağlayacağı ön görüldü. Üzerinde Linux tabanlı işletim sistemine sahip olması, bağımsız olarak çalışabilmesi, birçok çevre bileşenini üzerinde bulundurması ya da yazılımsal olarak desteklemesi gibi sebepler ile bu sistem üzerinde çalışmalar gerçekleştirildi.

Şekil 3.1: Çalışmada Kullanılan Raspberry Pi ve Kamera Modülü

Çalışmamızda 2 adet ana yazılım modülü bulunmaktadır ve bu modüller farklı yazılım dillerinde geliştirilmiştir. İlki şarj istasyonunu simüle eden "charger" ismini vermiş olduğumuz yazılım modülüdür. Bu yazılım modülü de kendi içerisinde şarj istasyonunu simüle eden "chargerframework" ve OCPP tabanlı sunucu yazılımı olarak iki ayrı yazılımdan oluşmaktadır. Bu çalışmalar, Java programlama dili kullanılarak gerçekleştirilmiştir.

İkinci yazılım modülü ise Raspberry Pi üzerindeki kamera modülünü yöneten ve plaka tanıma algoritmalarına veri aktarımını sağlayan yazılım modülüdür. Bu işlevleri yerine getiren kod ise Python 3.9 versiyonu ile geliştirilmiştir.

3.2 Şarj İstasyonu Simülasyonu

Çalışmaya ilk olarak şarj istasyonunun temel görevlerini simüle eden yazılım parçalarının geliştirilmesi ile başlanmıştır. Bu çalışma için seçilmiş programlama dili Java olmuştur. Bu dilin seçilme sebepleri ise çapraz platform derleme desteği sunması, nesne yönelimli programlama konseptlerinin kolaylıkla şekilde uygulanabilir olması ve yazılım komponentleri bakımından sunduğu geniş destek olarak özetlenebilir.

Şarj istasyonunu simüle eden yazılımları iki ana başlıkta inceleyebiliriz.

3.2.1 OCPP Test Sunucusu

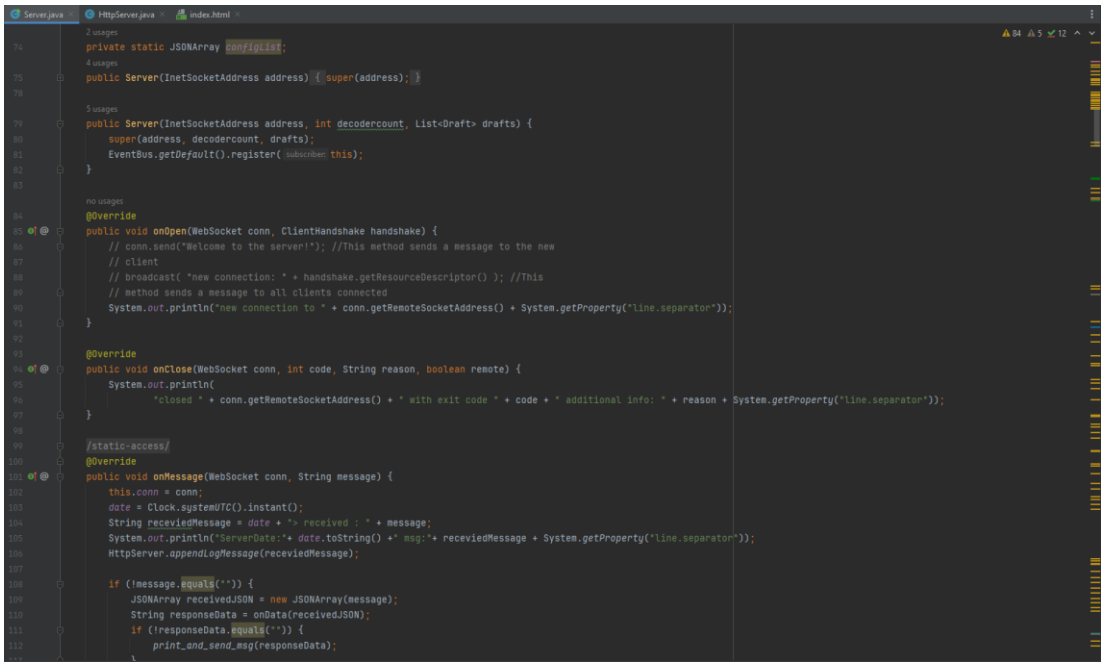
OCPP Test Sunucu yazılımının temel görevi OCPP formatına uygun şekilde haberleşen arka-uç yazılımını simüle etmesidir. Görevleri içerisinde gelen giden mesajların uygunluk kontrollerini yapmak, gerekli cevapların cihaza gönderilmesi ve Web arayüzünden cihaza komut iletimi sayılabilir. Bahsedilen görevleri yerine getirmek amacıyla yazılım, “Server” ve “httpServer” iki ana bölüme ayrılmıştır.

3.2.1.1 Sunucu Haberleşme Yazılımı

Sunucu haberleşme yazılımı, “Server.java” olarak isimlendirdiğimiz yazılım bloğudur. Http protokolü ile TCP/IP üzerinden cihaz ile haberleşmesinde görevlidir. OCPP formatında hazırlanmış komutları şarj cihazına gönderme ve gelen mesajların kontrolünü sağlayarak, uygun cevabı göndermek ana görevleri olarak nitelendirilebilir. Raspberry Pi cihazının IP adresini kullanarak 8081 portu üzerinden cihazla haberleşmeyi sağlamaktadır.

Şekil 3.2: “Server.java” Çalıştırılması ve Örnek Mesajlar

Yazılım içerisinde Java Sanal Makinesi (JVM) için geliştirilmiş web soket kütüphanesi üzerinden çalışmaktadır.

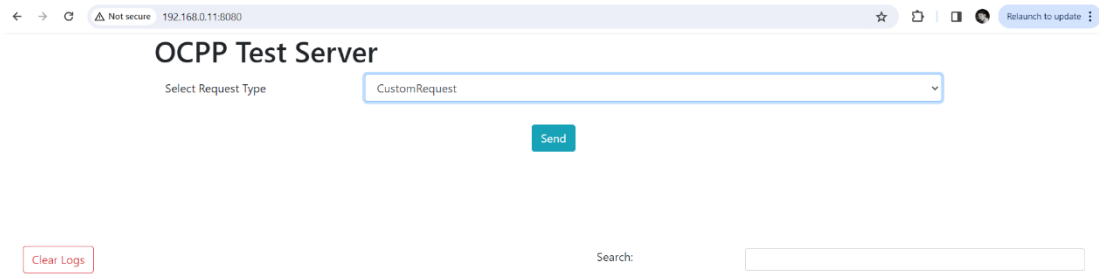


Şekil 3.3: “httpServer.java” İçerisinden Örnek Kod Parçası

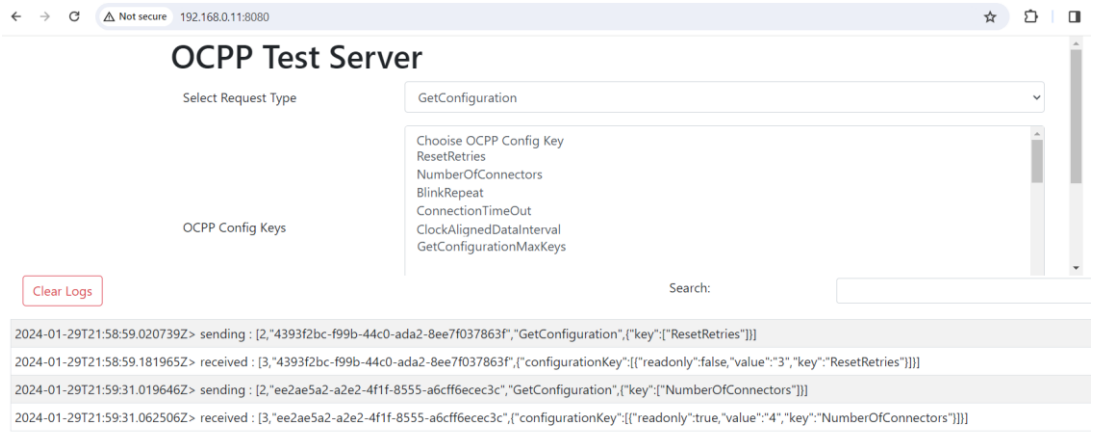
3.2.1.2 Sunucu Arayüz Yazılımı

İkinci yazılım parçası ise “httpServer.java” olarak adlandırdığımız arayüz yazılımıdır. Bu kütüphanenin görevleri ise internet tarayıcısı üzerinden cihaz ile iletişim kurulmasını sağlayan temel OCPP komutlarını seçerek gönderimini sağlamak ya da cihazdan gelen mesajları görüntüleyebilmektir.

Html dili ile ön yüz tasarımı yapılmıştır. Raspberry Pi cihazının IP’si üzerinden 8080 portunu kullanarak internet tarayıcısı üzerinden ulaşılabilmektedir.



Şekil 3.4: OCPP Test Sunucusu Arayüzü



Şekil 3.5: OCPP Test Sunucusu Üzerinden Gelen Giden Mesajların Görüntülenmesi

Yukarıdaki görselde de görülebileceği üzere, sunucuya gelen mesajlar “received”, sunucu üzerinden gönderilmiş mesajlar “sending” şeklinde etiketlenerek konsol ekranına basılmaktadır. Ayrıca yazılım içerisinde hazırlanmış bazı mesajlar OCPP haberleşme formatına uygun bir biçimde hazırlanıp istasyonu simüle eden yazılıma gönderilmektedir.

Görseldeki bir diğer bilgi ise OCPP veri formatıdır. Haberleşme mesajları JSON formatında oluşturulmaktadır. İlgili mesajını adı ile “transactionID” bilgi her mesajda bulunmaktadır. Bunlar haricinde iletilen mesaja göre farklı anahtar ya da değer bölümleri de mesaj içerisinde yer almaktadır.

```
2024-01-29T21:59:31.019646Z> sending : [2,"ee2ae5a2-a2e2-4f1f-8555-a6cff6ecec3c","GetConfiguration",{"key":["NumberOfConnectors"]}]
2024-01-29T21:59:31.062506Z> received : [3,"ee2ae5a2-a2e2-4f1f-8555-a6cff6ecec3c",{"configurationKey":[{"readonly":true,"value":"4","key":"NumberOfConnectors"}]}]
```

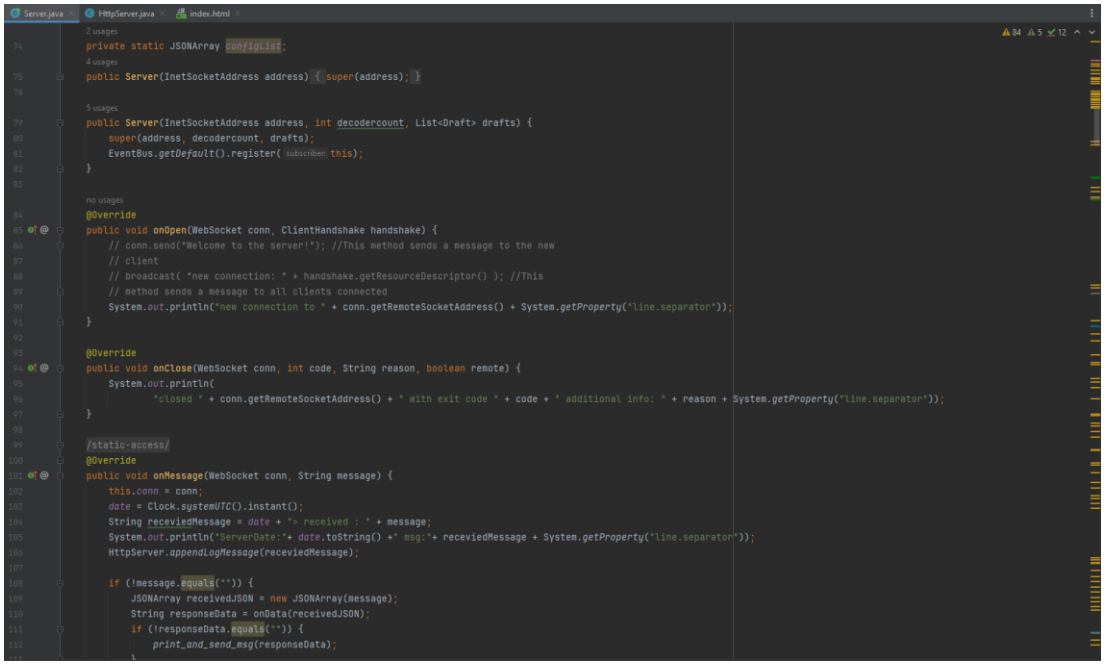
Şekil 3.6: Örnek bir veri formatı

Çalışmamızda sistemin diğer modülleri tarafından oluşturulan plaka bilgisini içeren bir şarj başlatma mesajını kontrol eden sunucu yazılımı bu modül içerisindedir.

3.2.2 Şarj Cihazını Simüle Eden Yazılım Modülü

OCPP sunucusu ile haberleşmeyi ve şarj işlemlerini kontrol eden yazılım modülünü geçmiş bölümlerde “chargerframework” olarak adlandırmıştık. Bu bölüm içerisinde ve sonrasında aynı isimle kullanılmaya devam edilecektir.

“chargerframework”, Java dili ile geliştirilmiş bir kontrol ve haberleşme yazılımı olarak özetlenebilir. Şarj istasyonunu simüle eden yazılım, içerisinde birden fazla görevde modül barındırmaktadır. Görevleri OCPP sunucusu ile haberleşmeyi sağlayıp, plaka tanıma sisteminden gelen bilgiyi ve şarj işleminin başlangıç ve bitişini kontrol etmektedir.



Şekil 3.7: OCPP Bağlantı Sınıfı Hakkında Örnek Kod

OCPP sunucusu ile bağlantı için “server.java” isimli yazılıma “client” olarak bağlanır. Bu soket üzerinden sunucu haberleşmesi gerçekleşmektedir. Sunucu haberleşmesinin uygun formatta yapılması için sunucu mesajlara ait sınıfları da içermektedir.

Şekil 3.8: Örnek OCPP Mesajı Sınıfı

Yazılım kendi içerisinde asenkron çalışmayı sağlamak için EventBus yapısını kullanmaktadır. Bu yapıdan ilerleyen bölümde daha detaylı bahsedilecektir.

Ayrıca modül Raspberry Pi içerisinde bir servis olarak sürekli çalışır vaziyette olacak şekilde tasarlanmıştır.

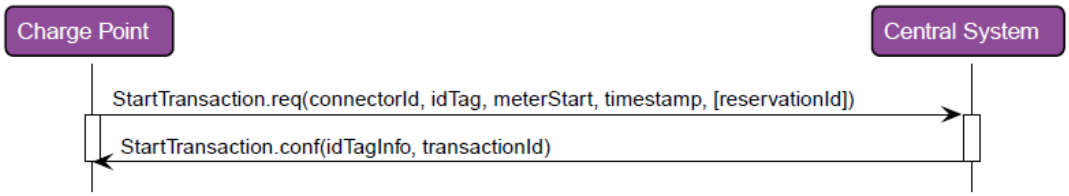
3.2.3 Şarj Başlatma ve Sonlandırma Prosedürleri

3.2.3.1 Şarj Başlatma Prosedürü

Şarj başlatma ve sonlandırma mesajları projede kullanılan en önemli mesaj tipleridir. OCPP 1.6 protokolünde detaylıca açıklanmış olan bu prosedür “starttransaction.req” ve “autrhorize.req” olarak iki mesajdan oluşmaktadır. Projenin temel unsuru olan plaka bilgisi “Authorize” isimli mesaj ile sunucuya aktarılacak olup, ”idTagInfo” cevap isimli mesaj ile sistemimizde kayıt bir araç olup olmadığı bilgisini istasyona geri döndürecektir. Örnek bir şarj başlatma prosedürü şekildeki gibidir. (OCPP 1.6, 2017)

2024-01-30T00:19:31.797279Z> received : [2,"09017e5f-ced5-45ea-9d32-2b50282b782f","Authorize",{"idTag":"1"}]
2024-01-30T00:19:31.797279Z> sending : [3,"09017e5f-ced5-45ea-9d32-2b50282b782f",{"idTagInfo":{"status":"Accepted"}}]
2024-01-30T00:19:39.269306Z> received : [2,"1048aff4-d21f-44b5-8ca3-868ac11af013","StartTransaction",{"connectorId":1,"idTag":"1","meterStart":451133,"timestamp":"2024-01-30T00:19:38.843Z"}]
2024-01-30T00:19:39.269306Z> sending : [3,"1048aff4-d21f-44b5-8ca3-868ac11af013",{"idTagInfo":{"status":"Accepted"},"transactionId":27}]

Şekil 3.9: Şarj Başlatma Mesajı

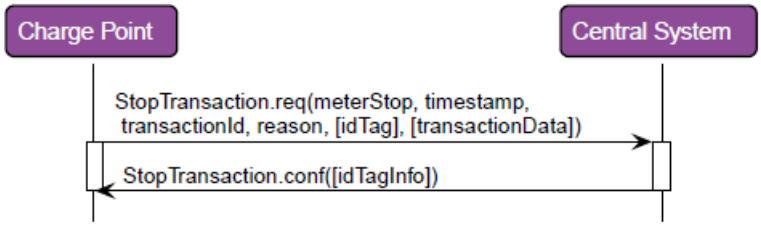


Şekil 3.10: Şarj Başlatma Prosedürü Sıralama Diyagramı

3.2.3.2 Şarj Sonlandırma Prosedürü

Bir şarj işleminin sonlandırılması istediğinde cihaz üzerinden yapılan bu girişim hakkında sunucu tarafının da bilgilendirilmesi gerekmektedir. Bu prosedürde kullanılan mesaj türü "stoptransaction" olarak isimlendirilmektedir.

İlgili mesaj içerisinde "idTag" olarak adlandırılan anahtar değerinin sistemin diğer bileşenleri tarafından okunan plaka bilgisi olması gerekmektedir. Bu sayede sunucu sistemi hangi şarjın sonlandırılmak istediğini teyit edebilecektir. Aynı anda birden fazla aracın şarj edildiği senaryoda bu bilgi önem arz etmektedir. Sunucu yazılımı mesaj içerisinde aldığı plaka bilgisi ile başlangıçta aldığı plaka bilgisini karşılaştırarak istasyona bir onay ya da ret cevabı göndermelidir.



Şekil 3.11: Şarj Sonlandırma Prosedürü Sıralama Diyagramı

Şekil 3.12: Şarj Sonlandırma Mesajı

3.2.4 Cihaz Üzerindeki Yazılım Modüllerinin Haberleşme Yöntemleri

Cihaz içerisinde yazılım modüllerinin kendi aralarında ya da modüller içerindeki thread veya kütüphaneler ile haberleşmek için kullandığı iki farklı yöntem bulunmaktadır. Java modüller içinde haberleşmek amacıyla EventBus haberleşme mimarisi kullanılmaktadır. Sistem içerisinde farklı yazılım modüllerinin birbiri ile haberleşmesi ise bir mesajlaşma servisi olan ZeroMq üzerinden sağlanmaktadır.

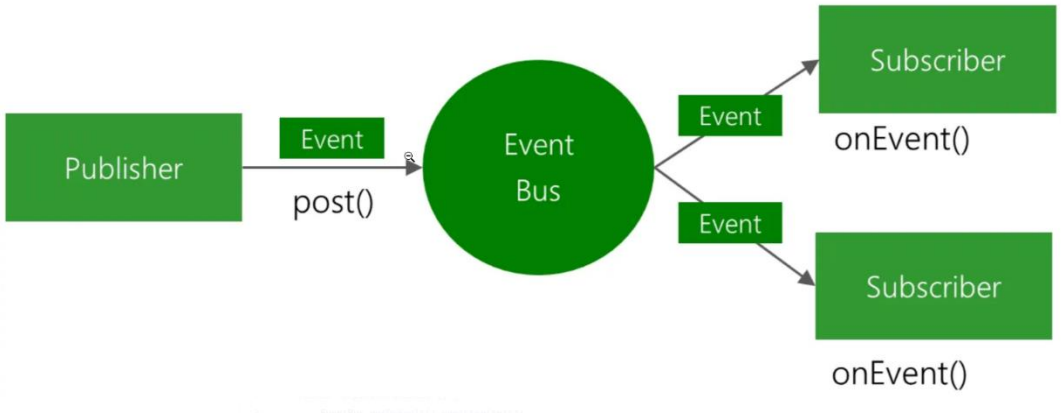
3.2.4.1 EventBus Haberleşme Modülü

"Event Bus" veya "Event-driven Architecture" (Olay Tabanlı Mimari) terimi, yazılım sistemlerinde olaylar (events) aracılığıyla iletişim kurma prensibini ifade eder. Bir Event Bus, bu tür bir mimariyi uygulamak için kullanılan bir bileşen veya hizmettir.

Java dilinde geliştirilmiş olan yazılımlarda threadlerin ve kütüphanelerin asenkron şekilde ve güvenli haberleşmesini sağlayan EventBus haberleşme mimarisidir.

Projemizde eventler genellikle sınıf formunda olup, önce sınıfınızı haberleşme hattına kaydedilmektedir. İlgili olayın Subscribe açıklaması kullanan yöntemleri tanımlanarak olay oluştuğunda uygun zamanda bu fonksiyon yardımıyla ilgili işlemler gerçekleştirilmektedir.

Şekil 3.13’ de çalışma prensibini özetleyen bir görsel yer almaktadır.



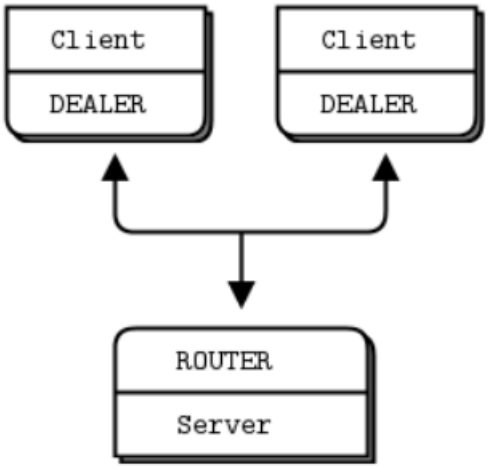
Şekil 3.13: EventBus Topolojisi

3.2.4.2 ZeroMq Kütüphanesinin Uygulanması

ZeroMQ, birçok programlama dilinde kullanılabilen bir mesaj geçişi (messaging) kütüphanesidir. ZeroMQ, dağıtık ve paralel sistemlerde iletişim kurmak için tasarlanmış hafif, esnek ve yüksek performanslı bir araçtır. İletişim kurmak isteyen uygulamalar arasında birçok iletişim modelini destekler ve bu nedenle çeşitli senaryolara uyarlanabilir.

Projede hız ve performans açısından avantajlı olması, farklı dil ve platformlara uygunluğu, aynı zamanda gelen mesajları kendi içerisinde sıralayarak işletebilmesi gibi avantajlarından faydalanmak amacıyla seçilmiştir.

Birçok farklı kullanım yöntemi ve deseni bulmaktadır. Projemizde yazılım modüllerinin hepsi Raspberry Pi üzerinde bulunduğundan “Router-Dealer” deseni ile çalışılmaktadır.

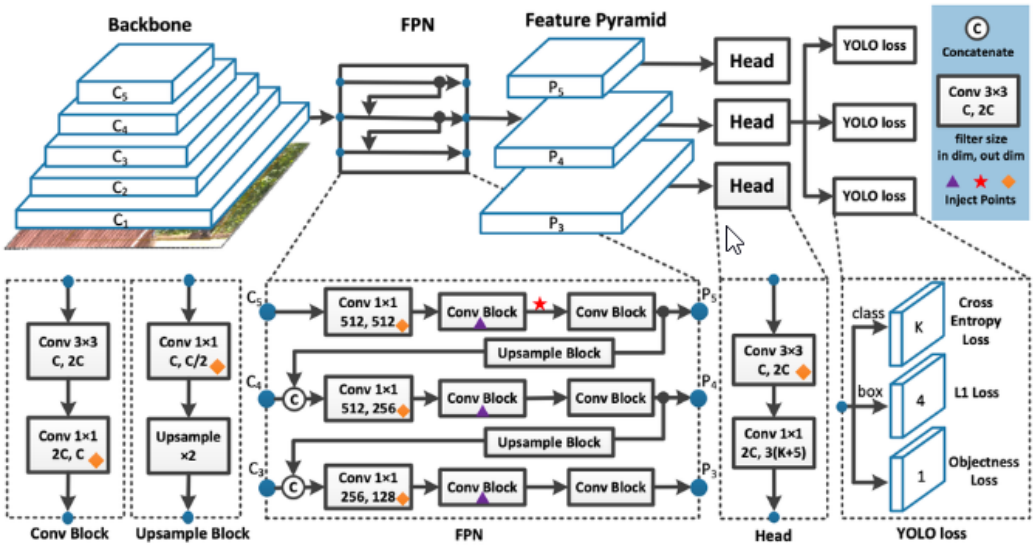


Şekil 3.14: ZeroMq Router Dealer Topolojisi

3.3 YOLO V7 ile Nesne Tanıma

3.3.1 YOLO V7 Derin Öğrenme Modülünün Kurulumu ve Kullanımı

YOLO V7 modeli, YOLO modelleri ailesinin son versiyonlardan biridir. YOLO modelleri tek aşamalı nesne dedektörü olarak adlandırılabilir. Bir YOLO modelinde, görüntü kareleri bir omurga(back-bone) aracılığıyla özelliklendirilir. Bu özellikler boyun içinde birleştirilip karıştırılır ve ardından ağın başına aktarılır YOLO, etrafına sınırlayıcı kutuların çizilmesi gereken nesnelerin konumlarını ve sınıflarını tahmin eder.



Şekil 3.15: YOLO V7 Mimarisi

YOLO, nihai tahminine ulaşmak için maksimum olmayan bastırma (NMS) yoluyla bir son işlem gerçekleştirmektedir.

YOLO modelleri küçük, çevik ve tek bir GPU üzerinde eğitilebilir oldukları için bilgisayarla görme ve makine öğrenimi toplulukları tarafından nesne algılamayı modellemek için yaygın olarak kullanılmaktadır. Küçük mimari, yeni makine öğrenimi çalışmacılarının YOLO'yu öğrenirken hızlanmalarını sağlarken, gerçek zamanlı çıkarım hızı da uygulayıcıların uygulamalarına güç sağlamak için minimum donanım hesaplaması tahsis etmelerine olanak tanıyor. Büyük mimarilerin aksine tüketici donanımında çalıştırılması daha kolay olması nedeniyle bu çalışma için uygun bulunmuştur. (Bochkovskiy, A., Wang, C, vd., 2022)

YOLO V7 modeli, WongKinYiu ve Alexey Bochkovskiy tarafından geliştirilmiştir. Bu çalışmaya açık kaynak olarak GitHub üzerinden ulaşılabilmektedir. Resmi kaynak koduna aşağıdaki komutu kullanarak ulaşılabilmektedir.

```
git clone https://github.com/WongKinYiu/yolov7.git
cd yolov7
```

Şekil 3.16: YOLO V7 Kaynak Kodu İndirme Yöntemi

YOLOV7 modelinin kaynak kodunu indirdiğimizde önceden eğitilmiş olduğundan bir ön-eğitim ağırlıkları ile gelmektedir. Bu ön eğitim Microsoft a ait COCO veri seti ile gerçekleştirilmiştir.

Standart hali ile modeli GoogleColab platformu üzerinden test görsellerimizden bir tane için çalıştırdığımızda sadece aracı tespit edebilmektedir.

Şekil 3.17: YOLO V7 Ön Eğitimli Hali ile Elde Edilen Sonuç

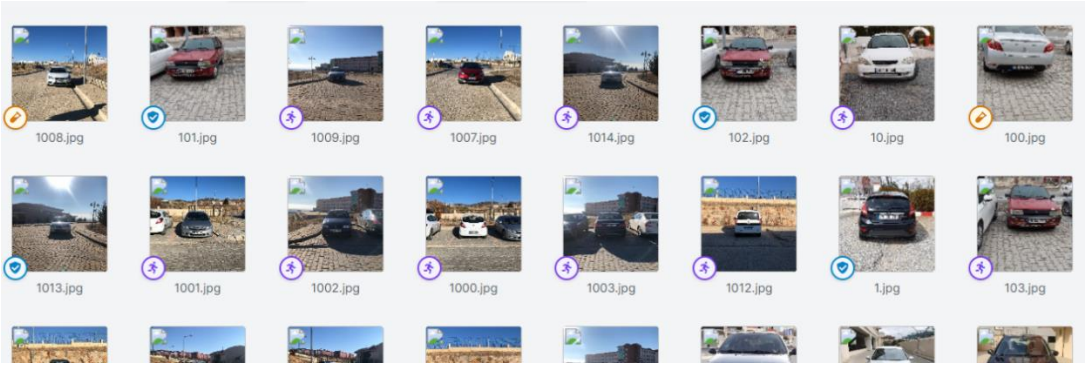
Bu sonucu değerlendirirsek, çalışmamıza uygun hale getirmek için modelin uygun plaka veri setiyle eğitilmesi gerekliliği ortaya çıkmaktadır.

3.3.2 Türkiye Plakaları Veri Seti

Bir önceki bölümde bahsedilen sebeple modelin eğitimi için uygun bir veri seti bulunmalı ya da oluşturulması gerekmektedir. Yapılan araştırma sonucunda Roboflow platformundan bir adet veri setine ulaşıldı.

Roboflow’ dan kısaca bahsetmek gerekirse bilgisayarlı görü işleme projelerini kolaylaştırmak ve hızlandırmak amacıyla kullanılan bir platformdur. Platform kullanıcılara görüntü verilerini yüklemeleri, etiketlemeleri ve işlemeleri için bir araç seti sağlar. Ayrıca, önceden eğitilmiş modelleri uygulamak ve kendi özel modellerinizi eğitmek için kullanabileceğiniz bir dizi araç ve kaynak sunmaktadır. Platform, endüstrilerde ve projelerde görüntü işleme ve nesne tespitiyle ilgilenen geliştiricilere, veri bilimcilere ve araştırmacılara hitap etmektedir.

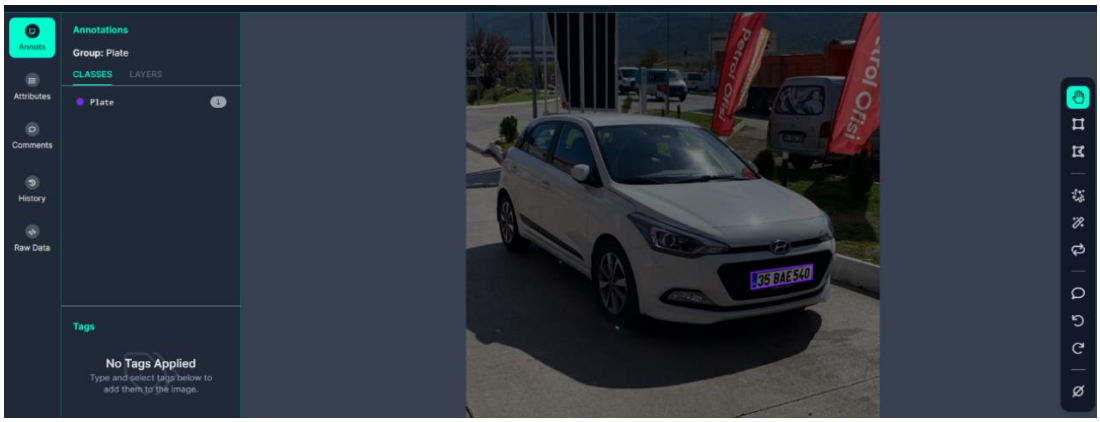
Bulunan veri seti platformdan indirilerek, kendi görsellerimizin de dahil edilmesiyle karma bir veri seti oluşturuldu. Bu veri setinde toplam 5484 adet görsel bulunmaktadır.



Şekil 3.18: Veri Setine Ait Görseller

Bu görsellerin 4857 tanesi eğitim seti, 419 tanesi doğrulama seti, 208 tanesi de test seti olarak belirlenmiştir.

Tekrar platforma yüklenen görseller üzerinde etiketleme çalışması yapılarak 419 adetlik doğrulama seti oluşturulmuştur.



Şekil 3.19: Veri Seti için Etiketleme Çalışması

3.3.3 YOLO V7 Modülünün Eğitilmesi

Modülün eğitilmesi için gerekli veri setinin hazırlanmasının ardından Google Colab platformu üzerinden çalışma yapıldı.

Öncelikle veri seti için bir "yaml" dosyası hazırlandı. Burada istenen kaç adet sınıflandırılacak görsel olduğu bilgisi ile bulunan nesnenin etiket bilgisi yer almaktadır.

```
train: ../train/images
val: ../valid/images
test: ../test/images

# Classes
nc: 1  # number of classes
names: ['0']  # class names
```

Şekil 3.20: Eğitim Seti için Hazırlanan Etiket Bilgisi

Hazırlık aşamalarının tamamlanması ile Colab platformunda eğitim başlatılmıştır. Kullandığımız Raspberry Pi kartının işlem gücünün böyle bir eğitim için çok yetersiz olması sebebiyle bu platform üzerinden işlem yapılmıştır.

Şekil 3.21: Veri Seti Eğitim Parametreleri

Eğitimde kullanılan parametreler ise yukarıdaki görselde görülmektedir.

- Batch (yığın) 16
- Epoch (Tekrar Sayısı) 55

Parametrelerden bahsetmemiz gerekirse, epoch (tekrar sayısı) bir eğitim veri setinin tamamının bir model tarafından bir kez geçirilmesini ifade eder. Eğitim süreci genellikle birden çok epoch boyunca gerçekleştirilir.

Bir modelin eğitilmesi, genellikle birçok iterasyon (tekrar) gerektirir. Her iterasyon, modelin ağırlıklarını güncellemek ve genel hata (kayıp) oranını azaltmak için kullanılır. Her bir iterasyon, bir veya daha fazla veri örneği içerir. Bir epoch, eğitim veri setindeki tüm örneklerin modele bir kez geçirilmesini ifade eder.

Epoch sayısının çok yüksek kullanılması aşırı uyumlamaya yol açabileceğinden, eğitim işleminin süresini göz önünde bulundurarak 55 tekrar sayısında karar kılınmıştır.

"Batch", bir eğitim algoritmasında kullanılan alt veri kümesini ifade eder. Bu küçük veri grupları, modelin ağırlıklarının bir güncelleme için kullanıldığı ve genel hata fonksiyonunun hesaplandığı bir eğitim iterasyonunu temsil eder.

Eğitim veri seti genellikle büyük ve birçok örnek içerebilir. Tüm veri setini aynı anda işlemek yerine, veriyi daha küçük ve yönetilebilir gruplara bölmek yaygın bir yaklaşımdır. Bu gruplar "batch" olarak adlandırılır.

Batch parametresi, bir eğitim sürecinde her bir iterasyonda kullanılacak olan örnek sayısını belirtir. Bu parametre, eğitim veri setini küçük parçalara böler ve her parçayı modele vererek ağırlıkların güncellenmesini sağlar. Daha büyük batch boyutları

genellikle daha hızlı eğitim sürelerine neden olabilir, ancak daha küçük batch boyutları, modelin daha genel ve genelleştirilebilir özellikler öğrenmesine yardımcı olabilir.

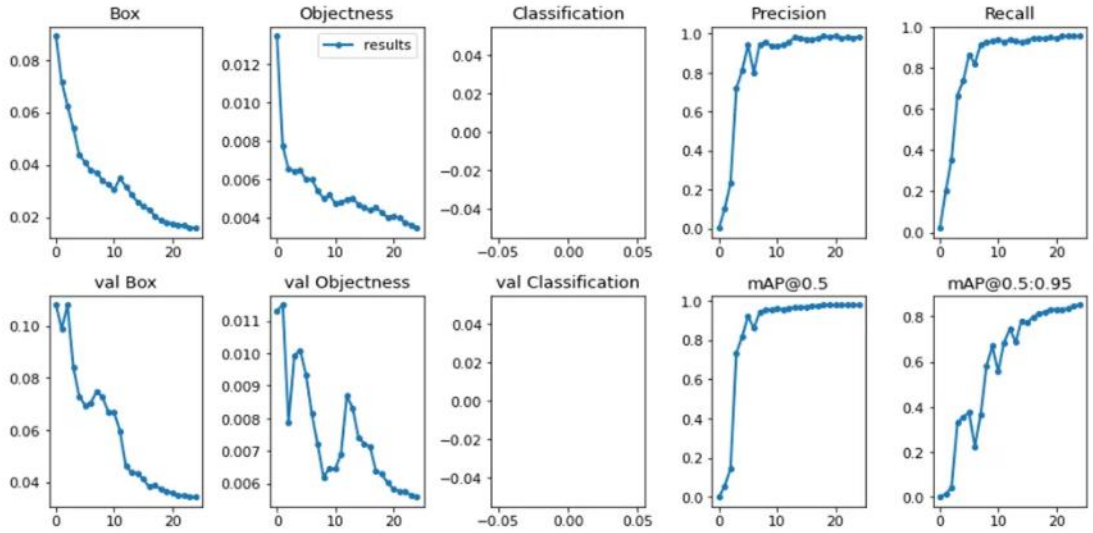
Eğitim sürecinde batch boyutu 16 olarak belirlendiğinden, her iterasyonda model 16 örnekle eğitilir ve ağırlıklar bu örneklerden türetilen gradyanlar kullanılarak güncellenir. Bu süreç, tüm veri seti üzerinde geçildiğinde bir epok oluşturur.

Model performansının etkilenmemesi açısından batch sayısı da 16 olarak belirlenmiştir.

```
autoanchor: Analyzing anchors... anchors/target = 3.12, Best Possible Recall (BPR) = 0.9988
Image sizes 640 train, 640 test
Using 2 dataloader workers
Logging results to runs/train/exp
Starting training for 55 epochs...

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      0/54     1.44G   0.06917  0.007293         0   0.07646        12       640: 100% 256/256 [04:34<00:00,  1.07s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 13/13 [00:15<00:00,  1.18s/it]
                 all         389         408       0.123      0.0858      0.0325     0.00526

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      1/54     13.6G   0.07693  0.003811         0   0.08074        12       640: 100% 256/256 [04:10<00:00,  1.02it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 13/13 [00:09<00:00,  1.43it/s]
                 all         389           0           0           0           0           0

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      2/54     11.5G   0.07086  0.004075         0   0.07494        13       640: 100% 256/256 [04:10<00:00,  1.02it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 13/13 [00:07<00:00,  1.64it/s]
                 all         389         408      0.0078       0.159     0.00141    0.000221

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      3/54     11.5G   0.07272  0.003162         0   0.07589         8       640: 100% 256/256 [04:05<00:00,  1.04it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 13/13 [00:09<00:00,  1.34it/s]
                 all         389           0           0           0           0           0

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      4/54     11.5G    0.0688  0.003222         0   0.07202         9       640: 100% 256/256 [04:01<00:00,  1.06it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 13/13 [00:08<00:00,  1.56it/s]
                 all         389         408       0.339       0.299       0.234      0.0708
```

Şekil 3.22: Modelin Eğitim Süreci

3.3.4 Modülün Eğitim Sonuçları



Şekil 3.23: Model Eğitim Sonuçları

Metrik	Değer
Ortalama Hassaslık	%99.3
Kesinlik	%97.7
Duyarlılık	%98.8

Tablo 3.1: Model Performans Metrikleri

Modelin eğitimi sonucunda ortaya çıkan değerler Şekil x' te görüldüğü gibidir. Grafiklere bakarak model eğitiminin başarılı olduğu yorumu yapılabilir.

Çalışmamızda modelin ana performansını belirleyen metrikler ise Tablo 1'de görülen değerler olarak kabul edilebilir. Nesneyi doğru ve hassas şekilde belirleyebilir olması bu çalışma kapsamında yeterli görülmektedir. Modelin eğitim metriklerine yakından bakacak olursak;

- Ortalama Hassaslık (Mean Average Precision):

  Nesne algılama modellerinin performansını ölçen bir metrik olarak karşımıza çıkar. Bu metrik, nesnelerin algılanması ve sınıflandırılmasındaki başarıyı değerlendirir ve 0 ile 1 arasında bir değer alır. Yüksek ortalama hassaslık değerleri, daha etkili bir performansı temsil etmektedir.

- Kesinlik (Precision):

Doğru pozitif tahminlerin toplam pozitif tahminlere oranını ifade eden bir metriktir. Bu metrik, yanlış pozitif tahminlerin etkilerini en aza indirmeye yöneliktir ve yüksek kesinlik, daha düşük yanlış pozitif oranı ile ilişkilidir.

- Duyarlılık (Recall):

Gerçek pozitif tahminlerin toplam gerçek pozitif durumlarına oranını ifade eder. Bu metrik, yanlış negatif tahminlerin etkilerini en aza indirmeye odaklanır ve yüksek duyarlılık, daha düşük yanlış negatif oranı ile bağlantılıdır.

Bu üç metrik ışığında modelin eğitim sonuçlarının yeterli olduğu sonucuna varabiliriz. Sonuçları görseller üzerinde de test ettiğimizde aşağıdaki sonuçlara ulaşmaktayız.

Şekil 3.24: Model Sonucu 1

Şekil 3.25: Model Sonucu 2

Şekil 3.26: Model Sonucu 3

Şekil 3.27: Model Sonucu 4

3.4 EasyOCR Yazılım Modülü ile Karakter Tanıma

3.4.1 EasyOCR Optik Karakter Tanıma Modülü Kurulumu ve Kullanımı

EasyOCR, kullanıcı dostu ve esnek bir Python dilinde Optik Karakter Tanıma (OCR) modülüdür. OCR teknolojisi, veri girişi otomasyonu ve görüntü analizi gibi bir dizi görevde kullanılır. Bilgisayarları fotoğraflardan veya taranmış belgelerden metin tanıma ve çıkarma yeteneği kazandırır.

Modül geliştiricilere, OCR veya bilgisayar görüşü alanında deneyimi olmayanlar dahil olmak üzere herkes için kullanımı kolay bir kütüphane sunar. Çoklu dil desteği, önceden eğitilmiş modeller ve hızlı kelime tanıma odaklı özellikleri ise güçlü yanlarındandır.

EasyOCR 80 adet dilde karakter tanıma yapabilmektedir. Ülkemizdeki plaka sisteminde latin alfabesine ait karakterler olmasına karşın, geniş ölçekte bir proje tasarlandığı takdirde geniş dil desteğinin faydalı olacağı düşünülmüştür.

Python geliştiricileri için EasyOCR, yazı tipleri ve metin düzenleri ile başa çıkma, doğruluk ve hız konusundaki odaklanması nedeniyle güvenilir bir seçenektir. Fotoğraflardan metin çıkarma sürecini basitleştirerek Python projelerinde kullanımını kolaylaştırmaktadır. Yukarıda bahsedilen birçok avantajı göz önünde bulundurularak projede kullanılmak üzere seçilmiştir.

Modül kullanımı için kurulum aşamaları şu şekildedir.

- Pytorch Kurulumu

Modülün çalışabilmesi için Pytorch kütüphanesinin yüklü olması gerekmektedir. Bu işlem için ilgili komut Şekil X' te görülmektedir.

```
pip3 install torch torchvision torchaudio
```

Şekil 3.28: EasyOCR için Pytorch Kurulum Komutu

- EasyOCR Kurulumu

Modülün kendisi ilgili komutlar ile kurulur. Komutun işletilmesinden sonra kullanıma hazır hale gelmektedir.

```
pip install easyocr
```

Şekil 3.29: EasyOCR için Kurulum Komutu

3.4.2 Görüntülerin Yazılıma Uygun Hale Getirilmesi

EasyOCR modülünden aldığımız sonucu iyileştirebilmek için YOLOV7 algoritmasından elde ettiğimiz plaka görüntülerini görüntü işleme uygulayabiliriz. Bu işlem için yapılabilecek birçok görüntü işleme metodu mevcuttur. Örneğin RGB'den gri tonlamaya dönüşüm, gürültü temizleme, erozyon ve genişletme, eşikleme, histogram eşitleme gibi yöntemler izlenebilir. Fakat projenin kısıtlı kaynakla çalışması sebebiyle RGB'den gri tona dönüştürme ve eğri düzeltme uygulanarak çalışma kararı verildi.

```python
import numpy as np
import cv2
from scipy.ndimage import rotate

# Skew Correction
def _find_score(arr, angle):
    data = rotate(arr, angle, reshape=False, order=0)
    hist = np.sum(data, axis=1)
    score = np.sum((hist[1:] - hist[:-1]) ** 2)
    return hist, score

def _find_angle(img, delta = 0.5,  limit = 10):
    angles = np.arange(-limit, limit+delta, delta)
    scores = []
    for angle in angles:
        hist, score = _find_score(img, angle)
        scores.append(score)
    best_score = max(scores)
    best_angle = angles[scores.index(best_score)]
    print(f'Best angle: {best_angle}')
    return best_angle

def correct_skew(img):
    # correctskew
    best_angle =_find_angle(img)
    data = rotate(img, best_angle, reshape=False, order=0)
    return data


def ocr_img_preprocess(img):
    # grayscale
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img = correct_skew(img)
    return img

```

Şekil 3.30: EasyOCR için Ön Görüntü İşleme Fonksiyonları

3.4.3 Karakter Tanıma Çalışması

Karakter tanıma için kod geliştirme işlemlerine başlandı. Oluşturulan sınıfın yapısı şu şekildedir. Öncelikle nesne algılama algoritmasından gelen çerçeve içine alınmış plaka görüntüsü okunur. Ardından görüntü işleme süreçleri tamamlanarak ilgili görüntü dizisi “ocr” kütüphanesine ait olan “readtext” metoduna gönderilir. Buradan dönen bilgi bir sözlük veri yapısına sahiptir. Bu veri okunan karakter dizisi ve güven bilgisini içermektedir.

Bu noktada artık okunan plakanın karakter dizisi elimizdedir. İlerleyen bölümde bu karakter dizisinin şarj istasyonu kontrol yazılımına iletilmesi kısmı incelenecektir.

```python
import easyocr
import numpy as np
import cv2
import time
import logging
from logging.handlers import RotatingFileHandler
from utils import ocr_img_preprocess


class EasyOcr():
    def __init__(self, lang = ['en'], allow_list = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ', min_size=50, log_level='INFO', log_dir = './logs/'):
        self.reader = easyocr.Reader(lang, gpu=False)
        self.allow_list = allow_list
        self.min_size = min_size
        self.log_level = log_level

    def run(self, detect_result_dict):
        if detect_result_dict['cropped_img'] is not None:
            t0 = time.time()
            img = detect_result_dict['cropped_img']
            img = ocr_img_preprocess(img)
            file_name = detect_result_dict.get('file_name')
            ocr_result = self.reader.readtext(img, allowlist = self.allow_list, min_size=self.min_size)#, paragraph="True"
            text = [x[1] for x in ocr_result]
            confid = [x[2] for x in ocr_result]
            text = "".join(text) if len(text) > 0 else None
            confid = np.round(np.mean(confid), 2) if len(confid) > 0 else None   #TBD
            t1 = time.time()
            print(f'Recognized number: {text}, conf.:{confid}.\nOCR total time: {(t1 - t0):.3f}s')

            return {'text': text, 'confid': confid}
        else:
            return {'text': None, 'confid': None}
```

Şekil 3.31: Karakter Tanıma Sınıfı Örnek Kod

Result

No.	Text	Confident Score
0	48 EJ 948	0.5962

Şekil 3.32: Karakter Tanıma Yazılımı Çıktısı

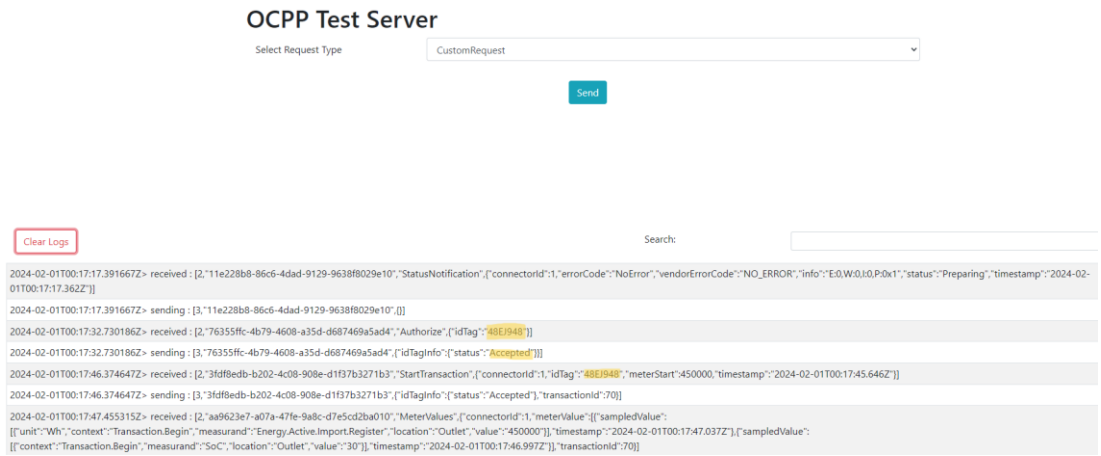
3.5 Karakter Tanıma Süreci Çıktılarının Şarj Kontrol ve Sunucu Yazılımlarına İletilmesi ve İşlenmesi

EasyOCR tarafından okunarak elde edilen plaka bilgisi, bir "string" türünde veri olarak elimizdedir. Bu noktadan sonra Python tarafında geliştirilmiş olan plaka tanıma sisteminin bu çıktıyı Java ile geliştirilmiş ayrıca bir servis olarak çalışan "chargerframework" isimli istasyon simülasyon yazılımına göndermesi gerekmektedir.

Daha önceki bölümlerde bahsetmiş olduğumu ZeroMq kütüphanesi bu noktada devreye girmektedir. Plaka verisi oluştuktan sonra birbirine ZeroMq kütüphanesi ile bağlı olan bu iki yazılım modülü arasında veri alışverişi gerçekleştirilmektedir.

ZeroMq üzerinden gönderen plaka tanıma modülü plaka bilgisi, "chargerframework" tarafından alınarak bir şarj başlatma sürecini tetikler. Bu noktada "chargerframework" şarj başlatma sürecindeki "Authorize" mesajında bulunan "idTag" anahtar kelimesinin karşılığı olarak plaka bilgisini işler. Daha sonra bu bilgi OCPP sunucu yazılımına gönderilir.

"Authorize" mesajı içerisinde bulunan plaka bilgisi sunucu tarafındaki plaka bilgileriyle karşılaştırılarak şarj başlatma sürecine ona ya da ret mesajı gönderilir.



Şekil 3.33: Plaka Bilgisi ile Şarj Başlatma İsteğine Sunucu Cevabı

Bölüm 4

Sonuçlar

Çalışma sonucunda yapılan geliştirmenin şarj başlatma süresini ciddi anlamda kısalacağı ön görülmektedir. Bu kısaltmanın sebebi ise diğer şarj başlatma yöntemlerine kıyasla işlem sıralarından farklı olarak aracın istasyona yanaşması ile yetkilendirme bilgisinin hazırlanmış olmasıdır. Bu sayede araç sahibi daha aracından inmeden istasyon şarj başlatmak için hazır konuma geçecektir. Diğer metotlarda ise araç kullanıcısının süreci başlatmak için yaptığı uygulamalarda geçen sürenin kazanıldığı görüldü.

Kullanıcı deneyimi ve işlem güvenliği açısından iyileştirmeler görüleceği düşünülmektedir. Diğer yöntemler kullanıcının yanında taşıma gereken kart ya da kredi kartı gibi nesnelere bağımlılığı ortadan kalkmaktadır. Şarj başlatma süresinin kısalmasının da kullanıcı memnuniyetini arttıracağı düşülmektedir.

Çalışmanın olumlu yanlarının yanı sıra geliştirilmesi gereken yönleri de mevcuttur. Bu çalışmada özellikle plaka tanıma sisteminde karakter tanıma yazılımında iyileştirmeler yapılması gelecek çalışmalar açısında faydalı olacaktır.

Bölüm 5

Gelecek Uygulamalar ve Olası Yan Çıktılar

Bu sistem sayesinde şarj istasyonu önüne park ederek gereksiz yere istasyonu meşgul eden araçların tespitinin yapılabilmesini sağlayacak altyapıyı içermektedir.

Bir başka uygulama alanı olarak, park alanı işgallerine karşılık bariyer ve benzeri mekanizmalar kurularak başka araçların park etmesinin de önüne geçecek mekanizmalar kurulması konusunda öncü bir çalışma olarak düşünülebilir.

Daha ileri uygulamalarda ise plaka bilgilerinin gerekli mercilere iletilmesi sağlanabilir. Bu sayede plaka üzerinden takip sağlaması konusunda güvenlik güçlerine yardımcı olacak bir sistem geliştirilebileceği ön görülmektedir.

Kaynaklar

Ahmad, A., Alam, M. S., & Chabaan, R. (2017). A Comprehensive Review of Wireless Charging Technologies for Electric Vehicles. IEEE Transactions on Transportation Electrification, 4(1), 38–63.

America EN, Phoenix A. (2012). Lessons Learned – The EV Project DC Fast Charge - Demand Charge Reduction. Prepared for the U.S. Department of Energy Award # DE-EE0002194.

Au, M. H., Liu, J. K., Fang, J., Jiang, Z. L., Susilo, W., & Zhou, J. (2017). A New Payment System for Enhancing Location Privacy of Electric Vehicles. IEEE Transactions on Industrial Informatics, 13(3), 1480–1488.

Bohn, T. (2019). Vehicle Charging; Low Level AC to DC Extreme Fast Charging For Commercial Vehicles, 1–6.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2022). YOLOv7: Trainable Bag-Of-Freebies Sets New State-Of-The-Art For Real-Time Object Detectors. Institute of Information Science, Academia Sinica, Taiwan.

Bräunl, T. (2012). EV Charging Standards, 1–5. Available from http://therevproject.com/doc/2012-EVcharging-s.pdf.

Channegowda, J., Pathipati, V. K., & Williamson, S. S. (2015). Comprehensive Review And Comparison Of DC Fast Charging Converter Topologies: Improving Electric Vehicle Plug-To-Wheels Efficiency. IEEE Int Symp Ind Electron, 263–268.

Dericioğlu, Ç., Yirik, E., Ünal, E., Cuma, M. U., Onur, B., & Tümay, M. (June 2019). A Review of Charging Technologies for Commercial Electric Vehicles. International Journal of Advanced Automotive Technology.

Ekici, Y. E., Dikmen, İ. C., Nurmuhammed, M., & Karadağ, T. (2021). A Review on Electric Vehicle Charging Systems and Current Status in Turkey. International Journal of Automotive Science and Technology, 5(4), 316–330. https://doi.org/10.30939/ijastech..958368.

Gao, Y., Zhang, X., Cheng, Q., Guo, B., & Yang, J. (2019). Classification and Review of the Charging Strategies for Commercial Lithium-Ion Batteries. IEEE Access, 7, 43511–43524.

Girshick, R. (2015). Fast R-CNN. Available at http://arxiv.org/abs/1504.08083.

Harighi, T., Bayindir, R., Padmanaban, S., Mihet-Popa, L., & Hossain, E. (2018). An Overview Of Energy Scenarios, Storage Systems And The İnfrastructure For Vehicle-To-Grid Technology. Energies, 11(8), 1–18.

He, J., Yang, H., Huang, H. J., & Tang, T. Q. (2018). Impacts Of Wireless Charging Lanes On Travel Time And Energy Consumption İn A Two-Lane Road System. Phys A Stat Mech its Appl, 500, 1–10. Available from https://doi.org/10.1016/j.physa.2018.02.074.

Karlsson, P., & Svensson, J. (2003). DC Bus Voltage Control for a Distributed Power System. IEEE Transactions on Power Electronics, 18(6), 1405–1412.

Liu, L., Kong, F., Liu, X., Peng, Y., & Wang, Q. (2015). A Review On Electric Vehicles İnteracting With Renewable Energy İn Smart Grid. Renewable and Sustainable Energy Reviews, 51, 648–661. Available from http://dx.doi.org/10.1016/j.rser.2015.06.036.

Mahajan, A. (2023). EasyOCR: A Comprehensive Guide. Available at https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168.

McPhail, D. (2014). Evaluation Of Ground Energy Storage Assisted Electric Vehicle DC Fast Charger For Demand Charge Reduction And Providing Demand Response. Renew Energy, 67, 103–108. Available from http://dx.doi.org/10.1016/j.renene.2013.11.023.

Mwasilu, F., Justo, J. J., Kim, E. K., Do, T. D., & Jung, J. W. (2014). Electric Vehicles And Smart Grid İnteraction: A Review On Vehicle To Grid And Renewable Energy Sources İntegration. Renewable and Sustainable Energy Reviews, 34, 501–516. Available from http://dx.doi.org/10.1016/j.rser.2014.03.031.

Ogunboye, O. E. (2023). Investigating A Road Traffıc Detection Using Yolov7. (Yayın No. 795193) [Yüksek Lisans Tezi, Üsküdar Üniversitesi]. YÖK Ulusal Merkezi. https://tez.yok.gov.tr/ulusaltezmerkezi/

Open Charge Point Protocol 1.5, (2012) https://www.slideshare.net/ganeshdks/ocpp-specification-15final

Open Charge Point Protocol 1.6, (2017) https://openchargealliance.org/protocols/open-charge-point-protocol/#OCPP1.6

Open Charge Point Protocol 2.0.1, (2020) https://openchargealliance.org/protocols/open-charge-point-protocol/#OCPP2.0.1

Pappas, J. C. K. (2021). A New Prescription for Electric Cars. Energy Law Journal, 35(1), 151–198. Available from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=96365764&site=ehost-live.

Pruthvi, T. V., Dutta, N., Bobba, P. B., & Vasudeva, B. S. (2017). Implementation of OCPP Protocol for Electric Vehicle Applications.

Reznichenko, M. (2023). Exploring payment options for EV charging: Which One İs The Best? https://www.linkedin.com/pulse/exploring-payment-options-ev-charging-which-one-best-max-reznichenko/.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January, 6517–6525.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. Available at http://arxiv.org/abs/1506.02640.

SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler J1772_201001. (2010).

SAE. (2017). Charging – What Can Be More Simple? https://www.sae.org/binaries/content/assets/cm/content/standards/chargingprimer.pdf

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep Neural Networks for Object Detection.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. Available at http://arxiv.org/abs/2207.02696.

You, X., & Zhao, C. (2017). Research and Implementation of OCPP 1.6 Protocol. Beijing University of Posts and Telecommunications.

Zheng, Z., Liu, T., Zhang, Y., & Cheng, X. (2011). Analysis on development trend of electric vehicle charging mode. ICEOE 2011 Int Conf Electron Optoelectron Proc, 1(Iceoe), 440–442.