**UYARILAR**

Bu dosya TÜRKİYE BİLİMSEL ve TEKNOLOJİK ARAŞTIRMA KURUMU, 1001 – BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA PROJELERİNİ DESTEKLEME PROGRAMI PROJE BAŞVURU FORMU temel alınarak hazırlanmıştır. Örnek başvuru evraklarına [1] link aracılığı ile ulaşılabilir. Proje yazımında dikkat edilmesi gereken hususlar ise şu şekilde sıralanmaktadır:

* Başvuru formunun Arial 9 yazı tipinde, her bir konu başlığı altında verilen açıklamalar göz önünde bulundurularak hazırlanması,
* Ekler hariç toplam 20 sayfayı geçmemesi beklenir.
* Değerlendirme projenin özgün değeri, yöntemi, yönetimi ve yaygın etkisi başlıkları altında yapılacaktır.

Daha fazla bilgi için [2] linkinden yararlanılabilir.

Ek bilgiler:

* Bölüm 3.1 de yer alan “Yönetim Düzeni: İş Paketleri (İP), Görev Dağılımı ve Süreleri” her bir iş paketinden sorumlu olan kişi en başa yazılır yardımcılar ise daha sonrasında belirtilir, Tez danışmanı sadece denetim ve kontrol aşamalarında yer alacaktır.

1. [**http://tubitak.gov.tr/tr/destekler/akademik/ulusal-destek-programlari/1001/icerik-proje-ornekleri**](http://tubitak.gov.tr/tr/destekler/akademik/ulusal-destek-programlari/1001/icerik-proje-ornekleri)

1. [**https://www.tubitak.gov.tr/tr/destekler/akademik/ulusal-destek-programlari/icerik-1001-bilimsel-veteknolojik-arastirma-projelerini-destekleme-pr.**](https://www.tubitak.gov.tr/tr/destekler/akademik/ulusal-destek-programlari/icerik-1001-bilimsel-ve-teknolojik-arastirma-projelerini-destekleme-pr)

**UYARILAR**

**Bu sayfayı Evrak teslimi esnasında**

**Siliniz**

**YILDIZ TEKNİK ÜNİVERSİTESİ**

**ELEKTRİK – ELEKTRONİK FAKÜLTESİ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME ÇALIŞMASI ARA RAPORU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bitirme Çalışması Konu Başlığı: Görüntü İşleme İle Gerçek Zamanlı Araç Yol Destek Sistemi Tarih: 16.11.2021** | | | |
| **Takım Üyeleri** | **Öğrenci Numarası** | **İsim Soyisim** | **İmza** |
| **1.** | **17014090** | **Barışcan Kurtkaya** |  |
| **2.** | **17014034** | **Arda Tezcan** |  |
| **3.** |  |  |  |
| **4.** |  |  |  |

# ÖZET

Türkçe ve İngilizce özetlerin projenin (a) özgün değeri, (b) yöntemi, (c) yönetimi ve (d) yaygın etkisinin ana hatlarını kapsaması beklenir. Her bir özet 450 kelime veya bir sayfa ile sınırlandırılmalıdır. Bu bölümün en son yazılması önerilir.

|  |
| --- |
| **Proje Özeti**  Günümüzde arabalar ulaşım için vazgeçilemez bir parça olmuşlardır ve neredeyse her yere bu araçlar sayesinde ulaşım sağlanmaktadır sadece ulaşım değil eşya, ürün nakliyatları için de hayatımızın vazgeçilmez parçalarından biri olmuşlardır fakat bu kadar önemli bir yeri olmasına rağmen tüm dünyada trafik kazaları yaşanmaya ne yazık ki devam etmektedir. Bu projede, görüntü işleme kullanılarak sürücünün yaptığı hareketleri izleyen ve şerit ihlali, dur trafik levhasını, yaya yolu gibi kurallara uymasını sağlayan bir sistem gerçeklenecektir. Bu sayede dikkatsizliğin ve şerit ihlali, başka bir araca yaklaşma sonucu oluşan ölümlerin, yaralanmaların ve kazaların azaltılması hedeflenmektedir. Geliştirilecek olan sistem öncelikle anlık olarak görüntüleri alıp bu görüntüleri gerektiğinde ön işlemeden geçiren ve sonrasında kullanılan evrişimli nöral ağa yönlendirerek görüntü içindeki şeritleri ve araçları tanıyan sonrasında tanıma işlemlerini başka bir nöral ağa yönlendirerek onları sınıflayan bu sınıflandırma sonucunda nesnelerin belli matematiksel işlemlerden geçirip konumunu öğrenen bu konuma göre uzaklık hesaplaması yapan bu işlemler sonucunda ara yüz yardımı ile bir geri dönütte bulunan program gerçeklenecektir. |
| **Anahtar Kelimeler: Otonom Araç, Sürücü Destek Sistemi, Bilgisayarlı görme, Nesne tanıma, Makine Öğrenmesi** |

# 1. AMAÇ VE HEDEFLER

*Amaç ve hedeflerinizde tasarım projesinde beyan edilen amaç ve hedeflerde değişiklik olduysa nedenleri ile açıklayınız.*

|  |
| --- |
| Projenin amacı, hem ürünlerin taşınması gerçekleştirilirken hem de kişilerin günlük yaşamlarında kullanmış olduğu, bir yerden bir yere gitmek için kullandığı araçların daha güvenilir olmasını sağlaması düşünülmektedir. Bu yapılan çalışmanın geliştirilmesi ile birlikte ölüm ve yaralanma sayılarının daha az olması hedeflenmiştir. Bu projede öncelikle sistemden alınan görüntülerin gerektiği taktirde bulunan en uygun ön işleme filtresinden geçirilmesi sonrasında bu görüntüdeki nesneleri doğru bir şekilde ve boyutta en hızlı şekilde tanınması daha sonrasında ise bu bulunan nesnelerin sınıflanarak hangi nesnenin ne olduğunu anlamak ve sınıflanmış bu nesnelerin konumlarını en optimal matematiksel yöntemi kullanarak hesaplayıp devamında bu duruma göre kullanıcıya gerçek zamanlı bir geri dönüş sunması hedeflenmektedir. Bu sayede trafikte yaşanan dikkatsizlik sonucu oluşmuş kazaların yapay zekâ sayesinde önüne geçerek azaltılması hedeflenmektedir. |

2.YÖNTEM

|  |
| --- |
| Görüntü işleme ile gerçek zamanlı araç yol destek sistemi için geliştirilecek olan sistemin blok diyagramı Şekil 1’de sunulmaktadır.              Önerilen sistem öncelikle görüntüyü alıp “Görüntü Ön İşleme” işlemine tabi tutacaktır. Sonrasında bir sonraki aşamaya geçerek nesneleri tanıyarak tanınan nesnelerin boyut ve alan gibi özelliklerini çıkartacaktır. Bir sonraki aşamada çıkartılan nesneler sınıflandırılarak türü tespit edilecektir. Bulunan ve sınıflandırılan bu nesnelerin uzaklıkları hesaplanarak sürücüye geri dönütte bulunulacaktır. Aşağıdaki alt başlıklarda, önerilen sistemin basamakları detaylı şekilde açıklanmıştır.  **2.1. Görüntünün Alınması**  Günümüzde gelişen teknolojiler sayesinde anlık olarak görüntü alınabilen kameralar bulunmaktadır. Bu kameralarda gerçek zamanlı akıcı bir görüntü elde edilmesi için gereken belli başlı özellikler aranmaktadır. Öncelikle görüntünün kaliteli olması için çözünürlük denilen bir kavram yer almaktadır. Bu kavram görüntü ekranda tanımlanırken kaç adet piksel ile oluşturulduğunu söylemektedir. Günümüz teknolojisinde 4k (3840 x 2160 piksel) görüntü kalitesine sahip hatta bundan daha yüksek çözünürlüklere sahip kameralar bulunmaktadır fakat bu teknoloji hala günümüzde pahalı olmasından kaynaklı çok fazla tercih edilmemektedir onun yerine 720 (1280 x 720 piksel) veya 1080 (1920 x 1080 piksel) görüntüler kaydeden cihazlar kullanılmaktadır. Bunun dışında görüntü için önemli olan bir başka şey ise en-boy oranıdır. Bu en boy oranı popüler olarak 16:9 ya da 4:3 formatında kullanılmaktadır ve yapılan bir araştırmaya göre insanlar geniş ekranları yani 16:9 en-boy oranı olan görüntüleri 4:3 olan görüntülerden daha fazla tercih etmektedir (Şekil 2) [4]. Farklarına gelecek olursak eğer 16:9 en-boy oranına sahip olan ekranlar 4:3 ile kıyaslandığında Şekil 3’teki gibi bir fark elde edilecektir bu da piksellerin yerleşme kısımlarını değiştirmektedir.    Yukarıdaki bahsedilen özellikler dışında görüntünün akıcı ve gerçek zamanlı olması için gereken en önemli özellik frame per second (FPS) yani saniye başına alınan görüntü anlamına gelmektedir. İnsan gözünün görüntülerin birleşimini video olarak görmesi için sektörde kameraların özelliklerine en az 30 FPS yani saniyede 30 adet fotoğraf çekecek şekilde ayarlamaktadır. Kaliteli görüntüler 60 FPS’te çekilmesi önerilmekte olup yavaş çekim görüntüler için 120-240 FPS ve üstü kullanılmaktadır. Bu sebeple sistemde görüntünün hem gerçek zamanlı olması hem de akıcı olması için 30 ya da 60 FPS görüntü alan bir kamera tercih edilecektir. Daha sonrasında bu alınan görüntüler gerektiği taktirde görüntü ön işleme işlemlerine tabi tutulacaktır. Görüntü Ön İşleme ile ilgili detaylı bilgi bir sonraki başlıkta detaylıca verilmiştir.  **2.2. Görüntü Ön İşleme**  Bilgisayarlar görüntüleri matrislerin içine yazılan sayılar ile tutarlar. Günümüzde renkli görüntülerde kullanılan renkler kırmızı, yeşil ve mavi olmak üzere 3 ana renkten oluşur. Görüntü işleme algoritmaları bu matris verilerini kullanarak belirli işlemler uygular ve nesne bulma, özellik çıkarımı gibi işlemler yapar fakat bazı algoritmalar renk sapmalarından ötürü gürültülere maruz kalırlar bu gürültüler yüzünden belli hesaplama hataları ve yanlış sonuçlar meydana gelir. Bu tarz hatalardan kurtulmak için “Görüntü Ön İşleme” yapılması gerekir. Aşağıda bazı görüntü ön işleme çalışmalarına yer verilmiştir. Aslında 2.3. Nesne Bulma ve Özellik Çıkartım İşlemi kısmında işlenecek olan YoloV3 uygulaması bu tarz gürültüleri engellemek için kendince bir sisteme sahiptir fakat ileride gerekmesi durumunda kullanılması için ayrı bir ön işleme kısmı oluşturulması düşünülmüştür. Şekil 4 ön işleme filtrelerinden geçirilmemiş kameradan alınan görüntüdür. Bu görüntüdeki her piksel 3 adet ışık renginden oluşan bir değer sonucu oluşmaktadır. Bu temel ışık renkleri sırasıyla Kırmızı (Red), Yeşil (Green) ve Mavi (Blue) renkleridir bunlara kısaca RGB renkler denilmektedir. Şekil 5’teki görüntü kameradan alınmış Şekil 4 görüntüsünün piksellerini Denklem 1’deki işlemler sayesinde Siyah- Beyaz renk düzlemi üzerindeki değerleri hesaplanıp oluşturulmuştur.        Şekil 6’da ise kameradan alınan görüntünün kırmızı, yeşil ve mavi renk değerleri alınarak 255 değeri üzerinden değilleri alınması ile oluşturulmuştur nesnelerin daha belirgin olmasını sağlamaktadır. Şekil 7’de ise kırmızı, yeşil ve mavi piksel değerlerinin belli bir matematiksel katsayı ile çarpılarak toplanıp belirlenen eşik değerini geçen yerleri beyaz geçmeyen kısımları siyah yapması ile oluşturulur. Verilen eşik değerinin çok dikkatli seçilmesi gerekir çok düşük veya çok yüksek eşik değeri verilmesi durumunda nesnelerin tespit edilmesi zorlaşır. Şekil 8’de ise Laplacian filtresi ile oluşturulmuş bir görüntü vardır şekillerin keskin noktalarını bularak bir çizim görüntüsü oluşturur nesnelerin ayırt edilmesini kolaylaştırır çok benzer diğer yöntem olan ve Şekil 9’da görülen Sobel Feldman filtresi ile çok küçük farklar barındırmaktadır. Görüntünün üstünde tuz-biber gürültüsü [6] olan ya da Gauss gürültüsü [7] adı verilen gürültüler ve bozukluklar olması taktirde sistem görüntüyü ön işlemeden geçirecektir. Ön işlemeden geçirdiği görüntüyü Yolov3 algoritması ile eğitilen CNN modeline gönderecektir. Yolov3 ve CNN modelleri bir sonraki başlıkta detaylıca açıklanmıştır.  **2.3. Nesne Bulma ve Özellik Çıkarım İşlemi**  Son yıllarda geliştirilen birçok nesne bulma ve özellik çıkartma algoritması bulunmaktadır. Bunlardan birkaçı Yolov3[8], Fast R-CNN[12], Faster R-CNN[13] ve Mask R-CNN[14], Yolov4[15] ve RetinaNet[19] gibi algoritmalardır. Bu algoritmalar kendi içinde yaptığı işlemler ile nesneleri tespit eder ve onların genişliklerini ve boyutlarını hesaplarlar. Bu hesapları yaptıktan sonra kendi önceden eğitilmiş olan nöral ağından geçirip nesneleri tanır. Bu çalışmada sürücüye gerçek zamanlı geri dönütte bulunabilmesi için Yolov3 algoritması kullanılması uygun görülmüştür. Şekil 10’da da görüldüğü gibi yapılan araştırmalara göre Yolov3 (You Only Look Once) algoritması diğer nesne tanıma ve özellik çıkartma uygulamalarına göre daha hızlı bir şekilde çalışmaktadır.  Yukarıdaki çalışma Microsoft’un geliştirmiş olduğu Coco veri seti ile eğitilmiş model ile denenmiştir. Yukarıdaki şekilde de görüldüğü gibi Yolov3 algoritması 50ms’nin altında görüntü işlemesi ile neredeyse gerçek zamanlı çalışmaktadır.  1 saniyedeki çıkarılan görüntü sayısı ≥ 1000 = 20 [8] 50  Yolo Algoritması ızgara sistemi ile çalışmaktadır yani öncelikle alınan görüntüyü S x S adet ızgaraya böler. Sonrasında önceden eğitilmiş olan evrilebilen sinir ağı sayesinde nesnenin merkezinin hangi ızgarada olduğunu bulur ve o nesnenin boyutunu bulmaktan o ızgarayı sorumlu tutar. Her ızgara sınırlayıcı kutular ve doğruluk ihtimali için bir tahminde bulunur. Bu tahminlere doğruluk puanları denir. Doğruluk puanları tahmin edilen kutunun içinde bir nesne olup olmadığını ve çizilen yeni kutunun nesneyi doğru kapsayıp kapsamadığını belirtir. Literatürdeki çalışmalarda bu değer Pr(Object) \* IOUtruthpred olarak tanımlanır. Eğer nesne yoksa bu değer sıfır olmalıdır olmadığı taktirde hatalar oluşur. Amaç burada tahmin edilen kutu ile gerçek nesnenin oranının (IOU) eşit yani 1 olmasını isteriz. Ayrıca her sınırlayıcı kutunun 5 adet tahmin değeri vardır. Bunlar x,y,w,h ve güven yani IOU değeridir. X ve Y ızgara hücresinin sınırlarına göre kutunun merkezini temsil eder. W ve H yani genişlik (width) ve yükseklik (height) ise görüntünün tamamına göre tahmin edilir. Güven yani IOU değeri ise nesnenin kendisi ile tahmin edilen kutunun ne kadar örtüştüğü ile hesaplanır. [9]    Evrişimli nöral ağlar [18] günümüz teknolojisinin bu kadar hızlı ilerlemesini sağlayan en önemli yapı taşlarından biridir. Nöral ağ yapılarına örnek olarak Darknet53, Darknet19, ResNet-101, ResNet-152 gibi yapılar bulunmaktadır. Bu yapıların mantığı şu şekildedir resimdeki piksel görüntülerini alıp modelin içindeki yapıya göre filtrelerden (3 ∗ 3, 7 ∗ 7, 2 ∗ 2) geçer. Darknet53 gibi modellerde 1 ∗ 1 ‘lik filtrelerde bulunmaktadır bunun sebebi kendinden önceki gelen özellik alanını azaltmayı amaçlamasındandır. Şekil 13’te görülen yapı darknet53 yapısıdır toplamda 24 evrişimli katman ve 2 adet tamamen bağlı katmandan oluşmaktadır. Darknet53 modeli Yolov3’te de kullanılmaktadır. [9] [10] Nesne tanıma ve özellik çıkartımı sonrası alınan bilgiler bir sonraki aşama olan 2.4. Sınıflandırma kısmına iletilmektedir.  **2.4. Sınıflandırma**  Sınıflandırmayı yapan kısımda ise bir nöral sinir ağı kullanılmaktadır. Yolov3’ün de kendi nöral sinir ağları bulunmakta fakat yapılacak olan projede yeni bir modelin sıfırdan eğitilmesi tasarlanmaktadır. Nöral yapılar aşağıdaki matematiksel modele göre çalışmaktadır:    Şeklinde çalışmaktadır. Buradaki 𝛼 bağlanan nöronu temsil etmektedir. 𝑓 işareti ise sigmoid ya da son dönemlerde popüler olan ReLU fonksiyonunu temsil etmektedir. 𝑤 değerleri sonuç olacak çıkacak nörona önceki katmandan bağlı olan nöronların o nörona olan ağırlığını yani ne kadar etki ettiğini temsil etmektedir. En sondaki 𝑏 değeri ise bias’ı yani başlangıç koşullarını ifade etmektedir. Şekil 14’te evrişimli bir sinir ağı resmedilmiştir. Soldaki diyagram havuzlanmış düğümleri vurgulamaktadır, sağdaki diyagram ise tam indüklenmiş grafiktir. Evrişimli filtrelerin genişliği sırasıyla 3 ve 2’dir. Bu yöntem ile, üst katmanlarda olan küçük genişlikteki filtre ile giriş cümlesindeki kelimeleri birbirinden çok uzaklarda tutabilir. Şekil 15’te günümüzde kullanılmakta olan bazı popüler evrişimli ağ yapılarını eğiten veri tabanı resmedilmiştir. Yukarıdaki COCO veri tabanını göstermektedir toplamda 70’ten fazla nesneyi tanımaktadır. Bunların içinden kullanılması planlanan temel nesneler Araba, İnsan ve Dur işaretleri olacaktır. İhtiyaç olması durumunda diğer nesneler de sisteme eklenecektir.  Nöral ağlar katmanlardan oluşan karmaşık bir yapıdır. Her bir nöron kendisine gelen bilgiyi işleyerek sonraki katmana ya da çıkışa vermektedir. Bu projede çalışacak olan nöral ağların kısaca çalışma mantığını açıklamak gerekirse en üst katmana giriş değeri (görüntü,pikseller) uygulanacaktır. Bu değerleri alan nöron önceden eğitilmiş modeline uygun olarak çıkışa bir olasılık verecektir. (Örnek çıkış: %88.6 Araba, %35.2 İnsan, %0.6 Şerit ve bu çıktıdaki verilen eşik değerine göre en yüksek olan Araba çıktısı ekrana yazdırılacaktır.) Şekil 16’da El yazısını inceleyip tanımlayan nöral ağ resmedilmiştir. Nöral ağ öncelikle görüntüyü alıp 768 adet nörona göndermektedir sonrasında nöronların içindeki fonksiyondan geçerek ağırlıklarına göre sonraki nöronları etkilemektedirler sonraki katman 192 adet nörondan oluşmaktadır bu nöronlar da kendi içindeki fonksiyona göre bir değer çıkarıp kendilerinden sonraki 30 adet nörona ağırlıkları doğrultusunda etki uygularlar ve sistem son 30 nörondan çıkan sonuçlara göre çıkış değeri olarak gösterilen 10 adet çıkış nöronuna etki uygulayarak doğru cevabı gösterir.    Bu aşamada elde edilen veriler mesafe hesaplama algoritmalarına gönderilecektir. Böylelikle nesnelerin konumu hesaplanacaktır. Bu konu hakkındaki ayrıntılı bilgiler alt başlıkta verilmiştir.  **2.5. Şeritler, Araçlar ve İnsanlar ile Arasındaki mesafeyi hesaplama ve Geri Dönütte Bulunma**  Bir sonraki aşamada sistem tanıdığı şeritleri, araçları ve insanları kendinden ne kadar uzak olduğuna dair bir modelleme yapılması beklenmektedir. Bulunan cisim ile kendi aracımız arasındaki piksel sayısı hesaplanarak resmin çözünürlüğüne göre tahmini bir uzaklık değeri tanımlayacaktır. Bu değer belli bir eşik değerinin altına düştüğünde sistem sürücüye uyarıda bulunacak veya simülasyonda aracı diğer şartlara uygun olarak diğer nesne ile arasındaki eşik değerini korumaya çalışacaktır.  Nesnenin uzaklığı hesaplanırken uzaklık bulma yöntemlerinden birinin kullanılması düşünülmektedir. Bunlar sırasıyla Hamming uzaklığı[20] [21], Öklid uzaklığı [22] [23] ve Manhattan uzaklığı [24] [25] diğer adıyla taxicab geometri olacaktır. Bu bahsedilen uzaklık ölçüleri 2 nokta arasındaki en kısa mesafeyi yani vektörü hesaplamaktadır.  Hamming uzaklığı iki nokta arasındaki uzaklığı hesaplarken kendi aracımızın konumu ile nesnenin konumlarının ikili sayı sistemi üzerindeki değerleri ile işlem yaparak uzaklık hesaplamaktadır. Pikseller aracılığıyla en kısa hangi yoldan gidileceğini bulmaktadır. Şekil 17’de görüldüğü gibi (Mavi) 010 ile 111 değerleri arasındaki en kısa yolu, (kırmızı) 100 ile 011 arasındaki en kısa yolu göstermektedir.    Öklid uzaklığı ise Pisagor yöntemini kullanmaktadır. P noktasından Q noktasına en kısa mesafeyi bulmak için Şekil 18’de görüldüğü gibi Denklem 3’teki adımları izlemektedir. Bu işlem ile Şekil-19’da benzer bir projede oluşturulmuş olan matrise benzer bir matris oluşturulması beklenmektedir.  Manhattan uzaklığı diğer adıyla taxicab uzaklığı adıyla da anılmaktadır. İsmi kare kare yapısında oluşan binalar arasında gezinen taksiye benzetilmesinden gelmektedir. İki nokta arasındaki kare kare yapıda olan en kısa yolu bulmaya yarar fakat Öklid’e göre en kısa yolu bulmuş sayılmaz Şekil 20’de görüldüğü gibi Öklid uzaklığı Manhattan uzaklığı göre daha kısa bir yol sunmaktadır. (Mavi: Manhattan uzaklığı Yeşil: Öklid uzaklığı) Bu sebepten dolayı projede Öklid uzaklığı kullanılması düşünülmektedir.  Bu kısımda öğrenilen nesne konumları oluşturulan takip mesafesi ve yavaşlama algoritmalarından geçerek sürücüye ara yüz üzerinden bir dönütte bulunacaktır. Ara yüz ile ilgili detaylı bilgi bir sonraki başlıkta verilmiştir.  **2.6. Sürücüye Geri Dönütte Bulunma**  En son kısım olan sürücüye geri dönütte bulunma kısmı için özel bir ara yüz tasarlanması düşünülmektedir. Bu ara yüzün içinde anlık görüntü, dur işareti ve test edilmesinin kolaylaşması için nesneler arasındaki uzaklıkların yazması düşünülmektedir. Örnek olarak Şekil 21’de gösterilen yapay zekâ ile çalışan arabalar üreten Tesla şirketinin ara yüzü gösterilmektedir. Ara yüzün tasarlanması için Java ya da Flutter dilleri düşünülmektedir.    Bu sayede sürücünün dikkatsiz olduğu durumlarda yayanın görülmemesi ya da diğer aracın yaklaştığının fark edilmemesi, dur işaretinin algılanmaması gibi sorunların ortadan kalkacağı düşünülmektedir. Bu dikkatsizliklerin ortadan kalkması sayesinde sürücü kusurlu kazalarda büyük bir düşüş yaşanması öngörülmektedir. |

# 3. BİTİRME PROJESİ ARA RAPOR TESLİM TARİHİNE KADAR YAPILAN İŞLER

Öncelikle dataset için yapılan araştırmanın sonucunda Oxford RobotCar Dataset’inde karar kılındı ve dataset’i edinmek için belli mailler atılıp dataset’e erişim sağlandı. Ardından bu dataset üzerinden “Çizgi” ve “Araba” nesneleri üzerine labellama işlemleri yapıldı ve Model eğitimine başlandı ancak bu eğitilmiş modelin test sonuçları başarısız olduğu için daha fazla data labellanmasına karar verildi. Ardından YoloV3 algoritmasının COCO dataseti ile eğitilmiş pretrained modelinin gece görüntülerinde başarısız olduğu tespit edildi. Bu sebeple gece ve gündüz için iki farklı model kullanılmasına karar kılındı. Bu gereklilikten dolayı dataset üzerinden araştırmalar yapıldı ve gece-gündüz farkını ayırabilecek yöntemler geliştirilmeye başlandı.



**4. PROJE YÖNETİMİ**

# 4.1.Yönetim Düzeni: İş Paketleri (İP), Görev Dağılımı ve Süreleri

*İş paketlerinde, görev dağılımında ve/veya iş tanımlarında bir değişiklik olduysa son halini belirtiniz.*

**İŞ-ZAMAN ÇİZELGESİ (\*)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **İP No** | **İş Paketlerinin Adı ve Tanımı** | **Kim(ler) Tarafından Gerçekleştirileceği** |  | |  |  |  | | |  |
| **Tasarım Proj** | | **esi** |  | **Bitirme Çalışması** | | |  |
| **Ekim** | **Kasım** | **Aralık** | **Ocak** | **Şubat** | **Mart** | **Nisan** | **Mayıs** |
| 1 | Veri Tabanı Oluşturma | Barışcan Kurtkaya, Arda Tezcan | X |  |  |  |  |  |  |  |
| 2 | Gece ve Gündüz sınıflandırması | Barışcan Kurtkaya, Arda Tezcan |  | X | x |  |  |  |  |  |
| 3 | Model Eğitimi | Barışcan Kurtkaya, Arda Tezcan |  |  | X | X | X |  |  |  |
| 4 | Sınıflandırma Algoritması | Barışcan Kurtkaya, Arda Tezcan |  |  | x | x | x |  |  |  |
| 5 | Uzaklık Hesaplama | Barışcan Kurtkaya, Arda Tezcan |  |  |  |  | x | x | x |  |
| 6 | Kullanıcıya Dönüt Tasarımı | Barışcan Kurtkaya, Arda Tezcan |  |  |  |  |  |  | x | x |
|  |  |  |  |  |  |  |  |  |  |  |

**(\*)***Ara rapora kadar kaydettiğiniz ilerlemeyi ve varsa yapmış olduğunuz değişiklikleri hesaba katarak tabloları doldurunuz.*



# 4.2.Başarı Ölçütleri

*Tasarım projesi yönetim planınızın başarı ölçütlerinde değişiklik olduysa, yapmış olduğunuz değişiklikleri hesaba katarak tabloları doldurunuz.*

# BAŞARI ÖLÇÜTLERİ TABLOSU (\*)

|  |  |  |  |
| --- | --- | --- | --- |
| **İP No** | **İş Paketi Hedefi** | **Başarı Ölçütü** | **Açıklamalar** |
| 1 | Veri Tabanı Oluşturma | 2 saatin üstünde görüntü elde edilmesi, %15 | Oxford RobotCar dataset’i ile başarılı bir sonuç alındı. |
| 2 | Gece ve Gündüz sınıflandırması | Gece ve Gündüz’ün %95 oranında ayrılması, %15 | Contrast farkı ile yapılan test sonuçlarında %97’nin üstündebir başarı alındı. |
| 3 | Model Eğitimi | Modelin nesneleri %95 üstünde tanıması, %30 | Darknet53 ve Resnet’in kullanılması düşünüldü |
| 4 | Sınıflandırma Algoritması | Nöral Ağın %95 nesneleri doğru bir şekilde sınıflandırması, %15 | YoloV3 ve FasterRcnn’in denenmesi planlandı. |
| 5 | Uzaklık Hesaplama | Algoritmanın nesne ile arasındaki mesafeyi %90 oranında doğru şekilde tahmin etmesi, %15 | Öklid ve Manhattan uzaklıklarından birisinin kullanılması düşünüldü. |
| 6 | Kullanıcıya Dönüt Tasarımı | Sürücüye gerekli bilgileri sade bir şekilde aktarmak, %10 | Kullanıcıya geri dönütün yapılması için ara yüz tasarımının yapılmasıdır. |

**(\*)** Tablodaki satırlar gerektiği kadar genişletilebilir ve çoğaltılabilir.

A picture containing text, clipart, window

Description automatically generated

4.3. Risk Yönetimi

Projenin başarısını olumsuz yönde etkileyebilecek riskler ve bu risklerle karşılaşıldığında projenin başarıyla yürütülmesini sağlamak için alınacak tedbirler (B Planı) ilgili iş paketleri belirtilerek ana hatlarıyla aşağıdaki Risk Yönetimi Tablosu’nda ifade edilir. B planlarının uygulanması projenin temel hedeflerinden sapmaya yol açmamalıdır.

# RİSK YÖNETİMİ TABLOSU (\*)

|  |  |  |
| --- | --- | --- |
| **İP No** | **En Önemli Riskler** | **Risk Yönetimi (B Planı)** |
| 3 | Görüntülerde çok fazla gürültü olması durumu. | Ön işleme kısmındaki bir filtre ya da filtrelerden geçirilmesi. |
| 4 | Yolov3 algoritmasının yeterince doğru bilgiler vermemesi. | Yolov3 algoritmasından Yolov4 algoritmasına geçilmesi. |
| 5 | Öklid uzaklığının doğru çalışmaması durumu. | Manhattan uzaklığının kullanılmaya başlanması. |

**(\*)** Tablodaki satırlar gerektiği kadar genişletilebilir ve çoğaltılabilir.

**KAYNAKLAR (ÖRNEK):**

[1] TÜİK Kurumsal. (2019). Retrieved 31 December 2020, from <https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2019-33628>

[2] National Highway Traffic Safety Administration fars.nhtsa.dot.gov/Main/index.aspx

[3] National Highway Traffic Safety Administration (2018). U Drive. U Text. U Pay. Retrieved 31 December 2020, from <https://www.nhtsa.gov/risky-driving/distracted-driving>

[4] Pitts, K., & Hurst, N. (1989). How much do people prefer widescreen (16\* 9) to standard NTSC (4\*

3)?. IEEE Transactions on Consumer Electronics, 35(3), 160-169.

[5] Saravanan, C. (2010, March). Color image to grayscale image conversion. In 2010 Second

International Conference on Computer Engineering and Applications (Vol. 2, pp. 196-199). IEEE.

[6] Azzeh, J., Zahran, B., & Alqadi, Z. (2018). Salt and Pepper Noise: Effects and Removal. JOIV:

International Journal on Informatics Visualization, 2(4), 252-256.

[7] Luisier, F., Blu, T., & Unser, M. (2010). Image denoising in mixed Poisson–Gaussian noise. IEEE

Transactions on image processing, 20(3), 696-708.

[8] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement.arXiv preprint arXiv:1804.02767.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. InProceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

A picture containing text, clipart, window

Description automatically generated

[10] Wang, H., Zhang, F., & Wang, L. (2020, January). Fruit Classification Model Based on Improved Darknet53 Convolutional Neural Network. In2020 International Conference on Intelligent

Transportation, Big Data & Smart City (ICITBS) (pp. 881-884). IEEE.

[11] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for

modelling sentences. arXiv preprint arXiv:1404.2188.

[12] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer

vision (pp. 1440-1448).

[13] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6),

1137-1149.

[14] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEEinternational conference on computer vision (pp. 2961-2969).

[15] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of

Object Detection. arXiv preprint arXiv:2004.10934.

[16] Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., ... & Hubbard, W. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE

Communications Magazine, 27(11), 41-46.

[17]Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE

conference on computer vision and pattern recognition (pp. 7263-7271).

[18] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint

arXiv:1511.08458.

[19] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object

detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[20] Hamming distance. (2020). Retrieved 31 December 2020, from <https://en.wikipedia.org/wiki/Hamming_distance>

[21] Norouzi, M., Fleet, D. J., & Salakhutdinov, R. R. (2012). Hamming distance metric

learning. Advances in neural information processing systems, 25, 1061-1069.

[22] Euclidean distance. (2020). Retrieved 31 December 2020, from <https://en.wikipedia.org/wiki/Euclidean_distance>

[23] Danielsson, P. E. (1980). Euclidean distance mapping.Computer Graphics and image processing, 14(3), 227-248.

[24] Taxicab geometry. (2020). Retrieved 31 December 2020, from

<https://en.wikipedia.org/wiki/Taxicab_geometry>

[25] Manhattan distance. (2019). Retrieved 31 December 2020, from

<https://xlinux.nist.gov/dads/HTML/manhattanDistance.html>

[26] Aryal, M. (2018). Object Detection, Classification, and Tracking for Autonomous Vehicle.

[27] Tesla Autopilot AI (2020). Retrieved 31 December 2020, from <https://www.tesla.com/autopilotAI>

A picture containing text, clipart, window

Description automatically generated

**BÜTÇE VE GEREKÇESİ (ÖRNEK)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Maki** | **ne ve Teçhizat Giderleri (\*) (06.1 + 06.3)** | |  |
| **Adı / Modeli** | **Alım Türü** | **Kullanım Gerekçesi** | **Bedeli (TL)** |
|  | Yurt İçi Yurt Dışı |  |  |
|  | Yurt İçi Yurt Dışı |  |  |
|  | Yurt İçi Yurt Dışı |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sarf Giderleri (\*) (03.2)** |  |  |
| **Adı** | **Alım Türü** | **Kullanım Gerekçesi** | **Bedeli (TL)** |
|  | Yurt İçi Yurt Dışı |  |  |
|  | Yurt İçi Yurt Dışı |  |  |
|  | Yurt İçi Yurt Dışı |  |  |