



**T.C.
YALOVA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**FEROMON TUZAKLARIN NEKTARİ GÜVESİNE KARŞI
DİJİTALLEŞTİRİLMESİ VE MEYVE BAHÇELERİNDE İLAÇLAMA
ZAMANINI TAHMİN ETMEYE YÖNELİK AKILLI TARIM UYGULAMASI**

YÜKSEK LİSANS TEZİ

Ersin TANRIKULU

DANIŞMAN: Dr. Öğr. Üyesi Güneş HARMAN

**YALOVA
Mayıs 2022**



T.C.
YALOVA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

FEROMON TUZAKLARIN NEKTARİ GÜVESİNE KARŞI
DİJİTALLEŞTİRİLMESİ VE MEYVE BAHÇELERİNDE İLAÇLAMA
ZAMANINI TAHMİN ETMEYE YÖNELİK AKILLI TARIM UYGULAMASI

YÜKSEK LİSANS TEZİ

ERSİN TANRIKULU
185105012

DANIŞMAN: Dr. Öğr. Üyesi Güneş HARMAN

YALOVA
MAYIS 2022

ETİK BEYAN

Yalova Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kuralları'na uygun olarak hazırladığım “FEROMON TUZAKLARIN NEKTARİ GÜVESİNE KARŞI DİJİTALLEŞTİRİLMESİ VE MEYVE BAHÇELERİNDE İLAÇLAMA ZAMANINI TAHMİN ETMEYE YÖNELİK AKILLI TARIM UYGULAMASI” başlıklı bu tez çalışmada; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksinin tespiti halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi taahhüt ve beyan ederim.

ERSİN TANRIKULU

ÖNSÖZ

Yüksek Lisans eğitimin boyunca akademik anlamda bana her zaman destek olan ve benden yardımlarını esirgemeyen sayın danışman hocam Dr. Öğr. Üyesi Güneş HARMAN' a teşekkür ederim.

Bu tez çalışmanın desteklenmesine olanak sağlayan Yalova Üniversitesi Bilimsel Araştırma Projeleri (BAP) Koordinasyon Birimine (Proje No: 2020/YL/0023) teşekkür ederim.

Mayıs-2022

Ersin TANRIKULU



İÇİNDEKİLER

ETİK BEYAN	i
ÖNSÖZ	ii
İÇİNDEKİLER.....	iii
SİMGELER ve KISALTMALAR LİSTESİ	v
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ	1
1.1. Derin öğrenmede GPU ların kullanımı	4
1.1.1. Tarih.....	4
1.1.2. Nesne Algılama Nedir?	5
1.2. Feromon Tuzağın Algılayacağı Zararlı Doğu Meyve Güvesi.....	5
1.2.1. Tanımı	5
1.2.2. Zarar Şekli, Yayılımı ve Ekonomik Önemi	6
1.2.3. Konakçıları	6
1.2.4. Mücadelesi	6
1.2.5. İlaçlama Yapılan Bölgede Yapılan Uygulamanın Değerlendirilmesi .	8
1.3. Tezin Amacı	8
2. NESNELERİN İNTERNETİ VE UYGULAMA ALANLARI	9
3. GÖRÜNTÜ İŞLEME.....	9
3.1. Kullanım Alanları	10
Görüntü İşlemenin Kullanım Alanlarından bazıları:	10
3.2. Görüntü İşleme Süreçleri	10
3.3. Görüntü İşleme Teknikleri	11
4. MATERYAL ve YÖNTEM.....	12
4.1. Derin Öğrenme	12
4.1.1. Tek Katmanlı Algılayıcılar.....	12
4.1.2. Çok Katmanlı Algılayıcılar	13
4.1.3. Eğitim.....	15
4.1.4. Evrişimsel Sinir Ağı.....	16
4.2. CUDA	19
4.3. YOLO	21

4.4.	Materyaller	28
4.1.1.	Bilgisayar	28
4.1.2.	Gömülü Kart.....	29
4.1.3.	GPRS Modülü	32
4.1.4.	Sıcaklık ve Nem Sensörü	32
4.2.	Veri Setinin Hazırlanması	33
4.2.1.	YOLO Veri Setinin Hazırlanması	33
4.3.	Model Oluşturma ve Eğitim.....	38
5.	BULGULAR VE TARTIŞMA.....	40
5.1.	Kullanılan Yöntemler	45
5.1.1.	Duyarlılık, Kesinlik, mAP ve F1 skor	45
5.1.2.	Kayıp Fonksiyonları	46
5.2.	Test Görselleri	48
6.	SONUÇLAR	51
	KAYNAKLAR	53
	ÖZGEÇMİŞ.....	55

SİMGELER ve KISALTMALAR LİSTESİ

KISALTMALAR

ALU	: Aritmetik Mantık Birimi (Arithmetic Logic Unit)
CUDA	: Compute Unified Device Architecture
CPU	: Merkezi İşlem Birimi (Central Processing Unit)
cuDNN	: CUDA Derin Sinir Ağı Kütüphanesi (CUDA Deep Neural Network Library)
DSA	: Derin Sinir Ağı (Deep Neural Network)
DVM	: Destek Vektör Makineleri (Support Vector Machine)
FCN	: Tam Evrişimsel Ağ (Fully Convolutional Network)
GPGPU	: Genel Amaçlı Grafik İşlem Birimi (General Purpose Graphics Processing Units)
GPU	: Grafik İşlem Birimi (Graphics Processing Unit)
IoU	: Birlik Üzerinde Kesişme (Intersection Over Union)
IOT	: Nesnelerin İnterneti (Internet of Things)
mAP	: Ortalama Kesinlik Değeri (Mean Average Precision)
MLP	: Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
ReLU	: Doğrultulmuş Lineer Birim Fonksiyonu (Rectified Linear Unit)
SGD	: Stokastik Gradyan İnişi (Stochastic Gradient Descent)
SIFT	: Boyut ile Değişmeyen Özellik Çıkarımı (Scale-Invariant Feature Transform)
SLP	: Tek Katmanlı Algılayıcı (Single Layer Perceptron)
SURF	: Hızlandırılmış Gürbüz Öznitelikler (Speeded Up Robust Feature)
YOLO	: Sadece Bir Kez Bak (You Only Look Once)
YSA	: Yapay Sinir Ağı (Artificial Neural Network)

TABLÖLAR LİSTESİ

Tablo 1. Bilgisayar Özellikleri	28
Tablo 2. Gömülü Kart Teknik Özellikleri.....	30
Tablo 3. Kamera Modülünün Teknik Özellikleri	31
Tablo 4. DHT22 Sıcaklık ve Nem Sensörü Teknik Özellikleri	33
Tablo 5. Karmaşıklık Matrisi	45



ŞEKİLLER LİSTESİ

Şekil 1. Görüntü İşleme Süreci. [12]	5
Şekil 2. Doğu Meyve Güveleri	6
Şekil 3. Tek Katmanlı Algılayıcı (TKA) Modeli	13
Şekil 4. Çok Katmanlı Algılayıcı	14
Şekil 5. Görüntü İşleme Süreci. [12]	17
Şekil 6. Görüntü İşleme Süreci. [12]	17
Şekil 7. Görüntü İşleme Süreci. [12]	18
Şekil 8. CUDA İşlem Akışı	19
Şekil 9. CPU ve GPU mimarilerinin kıyaslanması. [13]	20
Şekil 10. Darknet-53 Mimarisi. [15]	22
Şekil 11. Darknet Görüntü İşleme Süreci. [15]	22
Şekil 12. Nesnelerin Izgara Hücrelerinde Gösterimi	24
Şekil 13. Sınırlayıcı Kutu Örneği	25
Şekil 14. Tahmin edilen Sınırlayıcı Kutu ve Gerçek Sınırlayıcı Kutu	26
Şekil 15. Üçlü Teknik Gösterimi. [20]	26
Şekil 16. Raspberry Pi 4. [19]	29
Şekil 17. Raspberry Pi 4 Uyumlu Kamera Modülü	31
Şekil 18. Raspberry Pi GSM Shield	32
Şekil 19. Doğu Meyve Güveleri 2	34
Şekil 20. Roboflow İşlemleri	35
Şekil 21. Roboflow İşlemleri 2	35
Şekil 22. Etiketli Güve Resimleri	36
Şekil 23. Etiketli Güve Resimleri	37
Şekil 24. Etiketli Verinin Koordinatları.	37
Şekil 25. Eğitim Aşaması	39
Şekil 26. Eğitim Aşaması 2	39
Şekil 27. Eğitim Sonucu	40
Şekil 28. Yolov5s tensorboard sonuçları	41
Şekil 29. Yolov5s tensorboard sonuçları 2	42
Şekil 30. Yolov5s tensorboard sonuçları 3	42
Şekil 31. Yolov5s tensorboard sonuçları 4	43
Şekil 32. Yolov5s tensorboard sonuçları 5	43
Şekil 33. Yolov5s tensorboard sonuçları 6	43

Şekil 34. Yolov5s tensorboard sonuçları 7.....	44
Şekil 35. Yolov5s tensorboard sonuçları 8.....	44
Şekil 36. Güve Nesnelerinin Tespiti	48
Şekil 37. Güve Nesnelerinin Tespiti 2.	49
Şekil 38. Tespit Edilen Nesnelerin Doğruluk Oranları.....	49
Şekil 39. Sınıfların Tespiti	50



FEROMON TUZAKLARIN NEKTARİ GÜVESİNE KARŞI DİJİTALLEŞTİRİLMESİ VE MEYVE BAHÇELERİNDE İLAÇLAMA ZAMANINI TAHMİN ETMEYE YÖNELİK AKILLI TARIM UYGULAMASI

ÖZET

Tarım arazilerinde sebze ve meyve bahçelerinde zararlılarla mücadelede tabiata dost olan biyolojik, biyoteknik ve mekanik mücadele metotları kullanılmaktadır. Kullanılan metotlardan biri de feromon tuzaklarıdır. Bu tuzakların amacı etrafa yaydıkları karşı cinsin kokularıyla böceklerin yakalanmasını sağlamaktadır. Bu nedenle zirai mücadelenin ne zaman başlanması gerektiği konusunda zararlı popülasyon yoğunluğunun tespiti oldukça önemlidir. Ancak tarım arazisine gidilerek kurulan tuzakların sürekli olarak kontrolünün sağlanması gerekmektedir. Bu projede öncelikli olarak feromon tuzakların dijital hale dönüştürülmesi ve kullanılacağı tarım arazisine gidilmeden uzaktan takip edilebilirliğinin kontrolünün sağlanması amaçlanmıştır. Böylece kullanıldığı tarım arazisine göre zararlılarla zamanında mücadelenin yapılmasıdır. Bu çalışmada dijital feromon tuzak sistemini oluşturduktan sonra, seçilen bir meyve bahçesine, dijital feromon tuzağı yerleştirilmiştir. Dijital feromon tuzağı içerisinde yer alan raspberry pi 4, sıcaklık sensörü, nem sensörü, kamera ve GSM- GPS- GPRS geliştirme modülleri ile bahçenin sıcaklık, nem bilgileri ölçülmüştür. Bunun yanında tuzak içerisinden günlük fotoğraflar alınmış olup görüntü işleme merkezine aktarılmıştır. Elde edilen bu görüntüler üzerinde görüntü işleme teknikleri kullanılarak doğu meyve güvesi zararlılarının popülasyon yoğunluğu tespit edilmiştir. Buna ek olarak kullanılan sensörlerle elde edilen sıcaklık ve nem değerleri belirlenen risk eşiklerine göre SMS veya e-posta yolu ile bahçe sahibine uyarı mesajı olarak iletilmiştir. Böylelikle tarım arazisi sahibine ilaçlama zamanına ne zaman başlanacağı konusunda fırsat sağlanacaktır.

Anahtar Kelimeler: Görüntü işleme, Nesnelerin interneti, Raspberry pi, Feromon tuzak, Tarım

SMART FARMING APPLICATION FOR DIGITIZING PHEROMONE TRAPS AGAINST CYDIA MOLESTA MOTH AND DETERMINING POPULATION DENSITY OF HARMFULS

ABSTRACT

Biological, biomechanical and mechanical methods that are friendly to nature are used in the fight against pests in vegetable and fruit gardens in agricultural lands. Pheromone traps are one of the methods used. The purpose of these traps is to catch insects with the smells of the opposite sex they spread around. In this way, it is very important to determine the pest population density in terms of when to start agricultural control. However, the traps set up by going to the agricultural land should be constantly controlled. The aim of this project is to digitalize the pheromone traps that are primarily planned and to control their remote monitoring without going to the agricultural land where they will be used. The purpose of using this way, appropriate intervention is made in time with harmful insects according to the farmland used.. In this work , after creating a digital pheromone trap system, a digital pheromone trap will be placed in a selected orchard. The temperature and humidity information of the garden will be measured with raspberry pi 3, temperature sensor, humidity sensor, camera and GSM-GPS-GPRS development modules that will be included in the digital pheromone trap. In addition, daily photos will be taken from the trap and transferred to the image processing center. The population density of the harmful insects will be determined by using image processing techniques on these images. In addition, according to the determined risk thresholds the temperature and humidity values obtained by the sensors used will be sent to the owner of the farmland as a warning message via SMS or e-mail. It allows the user without going to the agricultural land or farmland to learn the density of harmful insects transmitted to the crop. In this way, it will provide an opportunity for the owner of the agricultural land to start spraying time .

Key Words: Image Processing, Internet of things, Raspberry pi, Pheromone traps, Agriculture

1. GİRİŞ

Tarımsal üretim sırasında ortaya çıkan zararlılarla yapılan kimyasal mücadelenin insan sağlığı ve çevre üzerine olumsuz etkilerinin anlaşılması ile birlikte, zirai mücadelede kullanılan kimyasal ilaçların azaltılması için birçok biyoteknik yöntem geliştirilmiş olup bunlardan biri de feromonların kullanılmasıdır.

Feromon, dişi böcekler tarafından, erkek böceğin dişiye bulması için salgılanan kimyasal maddelerdir. Bu maddelerin aynı zamanda sentetik olarak da üretimi yapılmaktadır.

Feromon Tuzaklar ise, içerisine feromon konulmuş bir kapsül ile kapsüle gelen zararlının yakalanmasını sağlayan yapışkan bir tabla ve tuzak gövdesinden oluşmaktadır.

Kimyasal mücadelede; ilaçlama zamanını belirlemek için orman ve meyve bahçelerinde kullanılmaktadır.

Yapılan literatür taramalarında, feromon tuzakların kullanımının avantajları ve dezavantajları yapılan araştırmalar ile anlatılmıştır. Teknolojinin gelişmesiyle birlikte feromon tuzaklarını dijitalleştirme çalışmaları hayata geçirilmiş ve uzaktan izleme, nesne sayımı, sıcaklık ölçümü uygulamalarında genellikle raspberry pi ve arduino aracılığıyla yazılım geliştirme ve ayrıca sensörler ve modüller kullanılmaya başlanmıştır.

2008 yılında Isparta Ormanlarında zararlı Lepidoptera ve Hymenoptera türleriyle mücadelede biyoteknik yöntemlerden biri olan delta tipi feromon tuzağıyla çalışma yapılmış, böcek popülasyonunun başarıyla izlenebildiği görülmüş ancak mücadele çalışmalarında sürekli orman arazilerine gidilip tuzakların kontrol edilmesi gerektiği için bu tuzakların kullanımının ekonomik ve pratik olmadığı görülmüştür [3]. Bu alanda yapılmış diğer çalışmalardan birinde Feromon Tuzakların uzaktan denetiminin yapılabilmesi konusunda; görünüm ve şekillerine göre, farklı tip ve ebattaki böcekleri tanımak için dört öznitelik çıkarma yöntemi uygulanmıştır. Böcekleri sınıflandırmak için öznitelikler k-en yakın komşu algoritması kullanılmıştır. Her böcek için, çeşitli öznitelik tanımlayıcı çoğunluk oylaması yöntemi füzyonlanarak sınıflandırma algoritmasına gönderilmiş ve uygulanan yaklaşımın performansı, tuzak ortamında toplanan böcek görüntüleri ile test edilmiştir. Eliptik Fourier tanımlayıcıları (EFD), Hu momentleri (Hu), Lokal ikili Örüntü (LBP) ve Radyal Mesafe Fonksiyonlar (RDF) isimli öznitelikler yönteminden LBP, böceklerin sınıflandırılmasında diğer üç yöntemden daha başarılı olmuştur. Gelecekte, makine öğrenme katmanını geliştirerek, veri kümesinin artırılarak ve gerçek zamanlı verilerle deneme yapılarak, sınıflandırma performansının arttırılabileceği hususunda tavsiyede bulunulmuştur [1]. Bir başka çalışmada ise zirai tuzakların içindeki böceklerin fotoğrafları çekilmiş ve çekilen fotoğrafların matlab programında görüntü işleme

teknikleri uygulanarak görüntüdeki böcek sayıları tespit edilmiştir [2]. Fatih ve arkadaşları çalışmalarında görüntü işleme tekniklerini kullanarak bir ortamda insan yoğunluğunu hesaplamaya çalışmış, görüntü işleme tekniklerinden morfolojik işlemler ve etiketleme işlemlerini uygulayarak her iki yöntemin de sonuçlarını karşılaştırmışlardır [12]. Yu Sun ve arkadaşları çalışmalarında Çin'de uzun süre ciddi şekilde çam türlerini tehdit eden, istilacı bir haşere olan *Dendroctonus valens* LeConte kırmızı terebentin böceği (RTB) ve Kırmızı terebentin böceğini (RTB) saymanın geleneksel feromon tabanlı manuel bir sayımla sadece zaman alıcı değil, aynı zamanda hataya da açık bir uygulama olduğunu düşündüklerinden, yetişkin RTB'sini bir sistemde saymak ve erken önlem için gömülü cihazlarda çalışan bir derin öğrenme detektörünün tasarlamışlardır [4]. Jin Kyo Yung ve arkadaşları çalışmalarında Kore'de göçmen bir böcek olan *Mythimna separata*'nın (Lepidoptera: Noctuidae) gerçek zamanlı izlenmesi için uzaktan algılamalı bir feromon tuzağı tasarlamışlardır. Tasarlanan bu sistem, zararlı için kullanılan feromon kapsülüne sahip modifiye edilmiş bir koni tuzağı, ışık kesintisine dayalı bir algılama modülü, kod bölümlü çoklu erişime dayalı bir sinyal iletim modülü, sistem kontrolü için bir ana elektronik kart, güneş kolektörüne dayalı bir güç kaynağından oluşmakta olup web sayfasına dayalı bir sinyal toplama ve görüntüleme sistemi arayüzü geliştirilmiştir. Uzaktan algılama tuzağındaki gerçek böcek sayısının sinyal sayısına oranı ($>92\%$), böceğin sinyal ritmi temelinde, yalnızca geceleri sınırlı bir süre içinde algılama, temelde sinyal duyarlılığının kontrolü ile iyileştirilmiştir. Algılama yazılımı programının ayarlanmasında sinyal iletimi için 1 saatlik aralık referans alınmıştır. Uzaktan algılama tuzağındaki sinyal oluşum örüntüsü, tuzaktaki yetişkin oluşumunun gerçek örüntüsüne kesin olarak benzediği gözlemlenmiştir (korelasyon katsayısı, > 0.98) [5]. Karlos ve arkadaşları çalışmalarında seralarda kurulu yapışkan tuzaklar üzerinde yetişkin evre beyaz sinek (*B. tabaci*) ve thrip (*F. occidentalis*) miktarının belirlenmesini amaçlamış olup görüntü işleme ve sınıflandırma yapay sinir ağları aracılığıyla kombinasyon şeklinde bir algoritma geliştirilerek zararlının erken tespiti ve sınıflandırılmasını sağlamışlardır. Deneysel sonuçlara göre geliştirilen algoritma örnek görüntülerden yetişkin beyaz sineğin tanımlanması için precision (0.96), recall (0.95) ve F-measure ile (0.95) değerleri elde edilmiştir. Tanımlama için benzer iyi performans yetişkin thrip için model precision 0,92, recall 0,96 ve F-measure için 0.94 değerleri elde edilmiştir [6]. Yu Lei ve arkadaşları çalışmalarında *Puccinia striiformis*'in havada taşınan üredinosporlarının kantitatif izlenmesi, buğday şerit pasının kontrol edilmesi için buğday tarlalarında kullanılan spor tuzak cihazların, stratejileri erken ve etkili bir şekilde geliştirmek için önemli bir süreç olduğunu, geleneksel mikroskopik spor sayma yöntemi esas olarak alındığında, sıkışmış sporların çok sayıda olması nedeniyle, bu yöntemin yoğun emek ve zaman

alıcı olduğunu ve sayma verimliliğinin düşük olduğunu, çıplak gözle gözlemlerin bazen çok büyük hatalara yol açtığını; alternatif bir yöntemin gerekli olduğunu tespit etmişler ve bu sebeple otomatik sayım için yeni bir algoritma geliştirmişler ve dijital görüntü işlemeye dayalı olarak pst üredinosporlarının tespitini ve sayımını yapmışlardır. [7]. Hüseyin ve arkadaşları çalışmalarında farklı açılarda yerleştirilen FV paneller ile Bilecik İlinde güneş enerjisinden elde edilecek maksimum gücün hangi eğim açılarında gerçekleştiğinin tespitini yapmak için Raspberry Pi aracılığıyla python programlama dili ve mysql veri tabanı kullanarak bir çalışma yapmışlardır [8]. Uğur ve arkadaşları çalışmalarında raspberry pi ile yaşlı ve engellilerin hayatını kolaylaştıracak akıllı bir ev uygulaması geliştirmişlerdir [9]. Ayşenur ve arkadaşları çalışmalarında Raspberry Pi 4 ile sensörler yardımı ile görüntü işleme teknikleri kullanarak Sürücü Yorgunluk ve Uyarı Sistemi yapmışlardır [10]. Ömer ve arkadaşları çalışmalarında LabVIEW ve Raspberry Pi 3 kullanarak gerçek bir sıcaklık ölçümü yapmışlardır. LabVIEW ve Raspberry Pi 3 arasında kablosuz haberleşme kurulmuş ve LabVIEW de programlama yapılmıştır. Raspberry Pi 3 ve DS18B20 sıcaklık sensörü ile sıcaklık ölçülmüş ve ölçülen değerler Raspberry Pi 3 ün wifi özelliği sayesinde LabVIEW programına aktarılmıştır [17]. Murat ve arkadaşları çalışmalarında yumurtanın tasniflenmesi ve paketlenmesi konusunda görüntü işleme tekniklerini kullanarak Raspberry Pi Tabanlı gerçek zamanlı bir uygulama yapmışlardır. Yapmış oldukları uygulama da ilk olarak temiz ve kirli yumurtalar ayırt edilmiş ikinci aşamada ise ebatlarına göre S, M, L ve XL olarak sınıflandırılmıştır [18]. Kenan ve arkadaşları Raspberry Pi ve nesnelerin interneti tabanlı sera takip sistemi geliştirmişlerdir. Geliştirilmiş sistemler sayesinde kablosuz algılayıcılardan alınan bir seraya ait iç ve dış sıcaklık, nem değerleri, hava kalitesi, ışık değerleri toprak nemi bilgileri kayıt altına alınmakta ve belirlenen alt ve üst limitleri aşıldığında kullanıcıya e-posta veya kısa mesaj (SMS) ile bilgilendirme yapılabilmesi sağlanmıştır [19]. Kemal ve arkadaşları çalışmalarında hasatı insan gücü ile zor bir şekilde yapılan kiraz meyvesinin hasatının daha kolay yapılabilmesi için görüntü işleme teknikleri kullanarak ağaçtaki kiraz meyvelerinin yerini tespit etmeye yönelik Raspberry Pi 3 tabanlı otonom bir sistem geliştirmişlerdir [20].

Makine öğrenmesinin en popüler alanlarından biri derin öğrenmedir ve Derin Sinir Ağını (DSA) kullanılmaktadır. DSA verilerin tanınabilir kavramlarını tespit etmeyi bilgisayarlara öğretmektedir. Genellikle görüntülerin ve videoların sınıflandırılması, doğal dil işleme, ses tanıma, bilgisayarlı görme vb. uygulamalarda DNA'lar kullanılmaktadır.

Görüntü işlemede DNA'ları kullanmak için, görüntülerde tüm piksellerin girdi veri kümesi olması gerekmektedir. Resim piksellerinden oluşan çok sayıda girdi veri kümesi, DNA'nın çok katmanlı yapısıyla da birleştğinde yüksek hesaplama gücü ihtiyacını doğurmaktadır. Grafik

işlem birimleri (GPU) yardımıyla oldukça karmaşık ve büyük DNA'ların eğitiminde, eğitim süresi aylar sürerken birkaç güne indirilmiş ve bu hesaplama maliyeti büyük oranda azaltılmıştır. Bunların başında ise CPU'ların donanımsal olarak gelişmelerinin yanı sıra, bu donanımı genel amaçlı programlamaya imkân tanıyan ve grafik işlemcilerin limitlerini ortadan kaldıran CUDA (Compute Unified Device Architecture) gelmektedir.

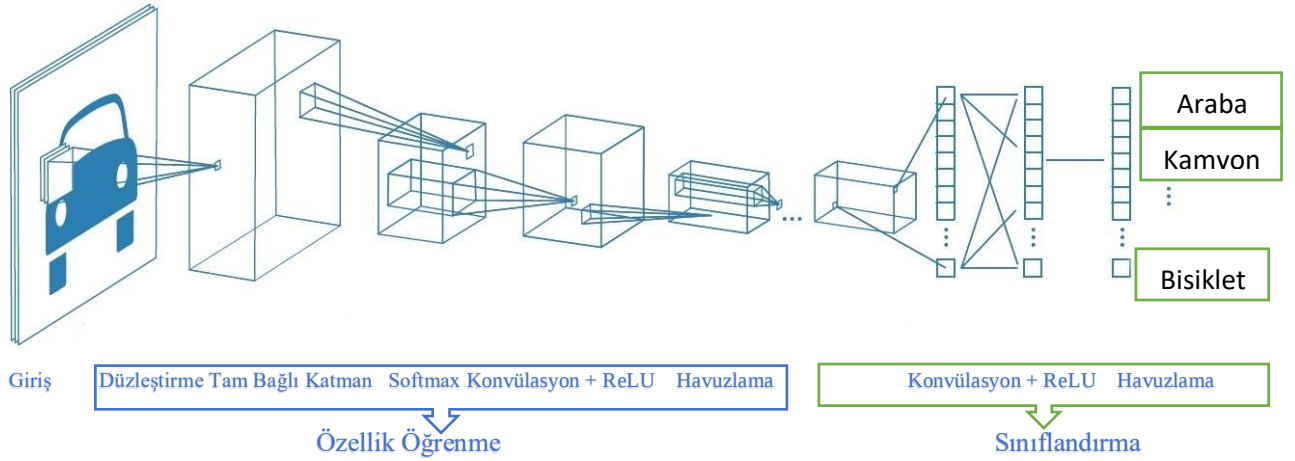
Bu çalışma ile Raspberry Pi 4 tabanlı oluşturulan, dijital feromon tuzağının içine yerleştirilmiş olan Raspberry Pi Kamera Modülü ile çekilen zararlı fotoğrafları üzerinde görüntü işleme teknikleri kullanılmış, şeftali bahçelerinde Doğu Meyve Güvesi Zararlısının popülasyon yoğunluğunun tespiti ve sıcaklık sensörü ile gün-derece hesabı yapılarak meyve için ilaçlama zamanının tahmini yapılmıştır. Proje sayesinde kendi elektriğini üretebilen, sıcaklık, nem bilgilerini ölçebilen ve tuzağın içerisindeki zararlı görüntülerini çekip görüntü işleme teknolojileri ile işleyebilen, kullanıcıya tarlaya gitmeden mahsulüne bulaşan zararlı yoğunluğunu iletebilen ve ilaçlamaya karar verme imkânı sunabilen akıllı tarım uygulaması geliştirilmiştir. Dijital feromon tuzağın getireceği büyük kolaylık yanında, çiftçiye zaman, iş gücü ve bilinçli ilaç kullanım konularında önemli tasarruf ve katkı sağlayacaktır. Ayrıca fazla kimyasal ilaç kullanımının önlenmesini sağlayarak; güvenilir, sağlıklı ve daha verimli gıda üretiminin yapılmasına büyük ölçüde yarar sağlayacaktır.

Tez çalışmasında eğitim süresini azaltmak ve daha başarılı sonuçlar elde etmek için, GPU ile CUDA Mimarisi kullanılmıştır.

1.1. Derin öğrenmede GPU ların kullanımı

1.1.1. Tarih

NVIDIA, GPU GeForce ilk ticari model olan 256'yı 1999'da piyasaya çıkmış ve 2000'li yılların ortasında grafik endüstrisini güçlendirmek için Grafik İşlem Birimi (GPU) teknolojisinin lider mucidi olarak kendini göstermeye başlamış, GPU oyunların grafik çerçevelerini CPU'lardan çok daha hızlı hale getirebilen paralel işlem yeteneğinden dolayı oyuncular arasında popülerlik kazanmaya başlamıştır. Böylece sorunsuz bir oyun deneyimi yaşatmıştır. 2007'de NVIDIA, NVIDIA GPU'larda GPU'da (GPGPU) Genel Amaçlı Hesaplama için CUDA API'den yararlanabilen yazılım programcıları için bir CUDA (Compute Unified Device Architecture) çerçevesi yayınlanmıştır.



Şekil 1. Görüntü İşleme Süreci. [12]

1.1.2. Nesne Algılama Nedir?

Nesne algılama, görüntülerdeki veya videolardaki nesnelerin örneklerini bulmak için bir bilgisayarlı görme tekniğidir. Nesne algılama algoritmaları, anlamlı sonuçlar üretmek için tipik olarak makine öğreniminden veya derin öğrenmeden yararlanır. İnsanlar resimlere veya videolara baktığında, ilgilendikleri nesneleri birkaç dakika içinde tanıyabilir ve bulabilirler. Nesne algılamanın amacı, bu zekâyı bir bilgisayar kullanarak çoğaltmaktır. Modern yöntemler iki ana tipte sınıflandırılabilir: tek aşamalı yöntemler ve iki aşamalı yöntemler. Tek aşamalı yöntemler çıkarım hızına öncelik verir ve örnek modeller, YOLO (You only look once) yani sadece bir kere bak RetinaNet'i içerir. İki aşamalı yöntemler, algılama doğruluğuna öncelik verir ve örnek modeller arasında Faster R-CNN (Faster Region Based Convolutional Networks) Daha Hızlı Bölge Tabanlı Evrişimli Ağlar, Mask R-CNN ve Cascade R-CNN bulunur.

En popüler kıyaslama MSCOCO veri kümesidir. Modeller tipik olarak Ortalama Ortalama Hassasiyet metriğine göre değerlendirilir.

1.2. Feromon Tuzağın Algılayacağı Zararlı Doğu Meyve Güvesi

1.2.1. Tanımı

Doğu Meyve Güvesi (Cydia Molesta Lep. Tortricidae) meyve ağaçları zararlısı olup kelebeğin kanat açıklığı 11-13 mm, arka kanatları gri pullara kaplı ve ön kanatları kahverengidir. Asıl konukçusu şeftali olup şeftali haricinde muşmula, ayva, elma ve armut da konukçuları arasındadır. Şekil 2 de feromon tuzağa yapışan doğu meyve güveleri gösterilmiştir.



Şekil 2. Doğu Meyve Güveleri

1.2.2. Zarar Şekli, Yayılışı ve Ekonomik Önemi

Doğu Meyve Güvesinin larvaları ağaçların sürgün ve meyvelerine zarar verir. Konukçusu olduğu ağaçların sürgünlerinin uç ve uca yakın bölgelerinden girerek beslenir. Sürgünler zarar gördükçe solarak devrilir. Larvalar sürgünleri terk ettikten sonra, sürgün uçları kurumaya başlar ve popülasyon yoğunluğu fazla olan bahçelerde bütün sürgünlerin uçları kuruduğu görülebilir. Meyvelerde ise meyvelerin birbirine temas ettiği yerlerden veya sap çukurlarından girerler. Meyveye giren larvalar doğrudan çekirdek kısmına doğru yönelerek çekirdek çevresindeki meyve etinden beslenir ve beslenmesini tamamladıktan sonra açmış olduğu delikten meyveyi terk eder. Meyvelerde oluşan yaralardan funguslar girerek meyvenin çürümesine neden olur. Marmara Bölgesinde İstanbul, Bursa, Kocaeli, Yalova ve Sakarya, Karadeniz Bölgesinde ise Amasya, Artvin, Giresun, Trabzon, Samsun ve Rize illerinde yaygın olarak bulunmaktadır.

1.2.3. Konakçıları

Doğu Meyve Güvesinin birinci dereceden konakladığı Şeftali ağaçlarıdır. Diğer meyvelerden ise, muşmula, ayva, badem, kayısı, elma ve armut ağaçları konakçılarıdır.

1.2.4. Mücadelesi

Kültürel Mücadele

Şeftali bahçelerinde başlangıç zararları sürgünlerde olmaktadır. Zarara uğrayan sürgünlerin tespit edilerek, haftada bir kez kesilmesi zararlı popülasyonunun azalmasına önemli ölçüde katkı sağlayacaktır.

Biyolojik Mücadele

Doğal düşmanların korunarak, etkinliklerinin artması için kimyasal mücadele yerine alternatif metotlara yönelim tercih edilmesi ve eğer kimyasal mücadele tercih edilecekse de doğal düşmanlara yan etkisi en az olan bitki koruma ürünlerinin tercih edilmesidir.

Kimyasal Mücadele

• İlaçlama Zamanı

Yeni kurulmuş şeftali bahçelerinde, sürgün zararı nedeniyle taç şekilleri bozulmakta olup sürgünlerde ilk girişler gözlemlendiğinde acilen ilaçlanmaya başlanması gerekmektedir. Meyve vermek üzere olan şeftali ağaçlarında ise bu sürgün zararı ihmal edilebilmektedir. Erkenci şeftali çeşitlerinde ise larvalar meyve hasadına kadar sürgünlerde beslendiğinde, meyvelerde zarar az görülmekte ve ilaçlamaya gerek kalmamaktadır. Ancak şeftalinin diğer çeşitlerinde feromon (eşeyssel çekici) tuzak kullanılarak, tuzak başına haftada 20 ve daha fazla kelebek yakalanması veya sürgün zararı %5' den fazla olduğunda, meyvelerdeki zararı önlemek için ilaçlama yapılması gerekmektedir.

Mart sonunda bahçeye feromon tuzakların asılması gerekir ve ilaçlamaya tuzaklarda ilk erginin yakalanmasından sonra sıcaklık toplamı (*EST*)¹ 400 gün-dereceye ulaştığında başlanıp, ilacın etki süresine göre ikinci ilaçlama yapılması gerekmektedir.

$$ETS^1 (gd) = \sum \left(\frac{\text{maxs} + \text{mins}}{2} - 10 \right) \quad [1]$$

gd: Gün-derece hesabını ifade etmektedir.

maxs: Maximum sıcaklığı ifade etmektedir.

mins: Minimum sıcaklığı ifade etmektedir.

• Kullanılan Bitki Koruma Ürünleri Ve Dozları

Tarım ve Orman Bakanlığı'nın yayınladığı Bitki Koruma Ürünleri kitabında tavsiye edilen ürünler ve dozlar kullanılmaktadır.

• Kullanılan Alet Ve Ekipmanlar

Bahçelerde yapılacak ilaçlamada, hidrolik bahçe pülverizatörü veya motorlu bahçe pülverizatörü kullanılmaktadır.

• İlaçlama Tekniği

Ağaçlar içten dışa doğru kuru yer kalmayana kadar ilaçlanmaktadır.

1.2.5. İlaçlama Yapılan Bölgede Yapılan Uygulamanın Değerlendirilmesi

İlaçlama sonrasında, 10 adet ağacın meyveleri hasat zamanı bulaşıklık ve zararlı yönünden sayılmaktadır. Zarar gören ve sağlam meyvelerin oranlanarak yapılan uygulamanın etkisi ortaya konulmaktadır. Zarar oranı eğer %2' den az ise uygulanan mücadelenin başarılı olduğu kabul edilmektedir.

1.3. Tezin Amacı

Yapılan bu çalışmada en önemli ve en temel amaç Feromon Tuzaklarının dijital hale dönüştürülmesidir. Bu sayede akıllı tarım uygulamaları geliştirilerek güvenilir, sağlıklı ve daha verimli gıda üretiminin yapılabilmesi amaçlanmıştır. Yapılan çalışma kullanıcı tarlaya gitmeden mahsulüne bulaşan zararlı yoğunluğunu tespit edebilen, sıcaklık ve nem ölçümü yapıp bu bilgileri kullanıcıya iletebilen ve ilaçlamaya karar verme imkânı sunabilen akıllı tarım uygulama sistemi ortaya koymuştur.

2. NESNELERİN İNTERNETİ VE UYGULAMA ALANLARI

Nesnelerin İnterneti nedir?

Nesnelerin İnterneti, herhangi bir cihazı internete ve diğer bağlı cihazlara bağlama anlamında kullanılmaktadır. Kısa olarak IoT olarak adlandırılmaktadır. IoT, “Internet of Things” kelimesinin kısaltmasıdır. Internet of Things (IoT), kullanım biçimleri hakkında ve çevrelerindeki ortamla ilgili veri toplayan ve paylaşan, birbirine bağlı nesneler ve insanlardan oluşan dev bir ağdan oluşmaktadır.[16]

IoT Uygulama Alanları

IoT birçok farklı alanda kullanılmakta olup günlük hayatı kolaylaştıran uygulamalar dışında, endüstriyel ve üretim alanlarında da büyük faydalar sağlayan uygulamaları yoğun bir şekilde kullanılmaktadır. Bu uygulamalar enerji, Telekom ve otomotiv dahil olmak üzere çok sayıda alanı kapsar.

Tüketici kapsamında kombi sistemleri, aydınlatma ve akıllı termostatlar gibi cihazlarla donatılmış akıllı evler akıllı telefonlar ve bilgisayarlar aracılığıyla kontrol edilmektedirler. Akıllı şehirler, akıllı mağazalar, sağlık sektöründe hastalar için oluşturulan verilerin izlenmesi, akıllı binalarda ve sokak lambalarında sıcaklığın kontrol edilerek enerji maliyetlerinin düşürülmesi, tarım alanında sulama sistemlerinin otomasyona bağlanmasında, bağlı sensörleri kullanarak tarlalarda ışık, toprak nemini izlemede ve birçok alanda kullanılmaktadır.

3. GÖRÜNTÜ İŞLEME

Dijital ortama aktarılan görseller üzerinde ihtiyaç duyulan bilgileri edinebilmek ve kaydedilmiş olan belirli görüntülerin işlenip mevcut grafik ve resimlerin değiştirilerek iyileştirilmesi için görüntü işleme yöntemi kullanılmaktadır. Örneğin, belgeler veya fotoğraflar taranıp dijital ortama iletilirken görüntü kalitesinde bozulmalar olmakta ve işte bu aşamada görüntü işleme yöntemi devreye girmektedir. Görselde yaşanan bozulmaları ve görüntü kalitesini en aza indirebilmek için görüntü işleme yöntemlerine ihtiyaç duyulmaktadır. İkinci bir örnek ise Google Earth aracılığı ile erişilebilen haritalardır. Uydu aracılığı ile alınan görüntüler, görüntü işleme teknikleri ile iyileştirilmekte ve böylece oldukça kaliteli görseller elde edilmektedir.

3.1. Kullanım Alanları

Görüntü İşlemenin Kullanım Alanlarından bazıları:

- Yapay Sinir Ağları (Artificial Neural Networks)
- Markov Rastgele Alanı (Markov Random Field)
- Dalgacık Dönüşümü (Wavelet Transformation)
- Yönlendirme Süzgeçleri (Routing Filters)
- İteratif Hücresel Resim İşleme Algoritması (Iterative Cellular Image Processing Algorithm)
- Görüntü İletimi (Image Transmission)
- Genetik Algoritma (Genetic Algorithm)
- Bulanık Mantık (Fuzzy Logic)

GÖRÜNTÜ İŞLEME NELERDE KULLANILIR?

Genel olarak görüntü işlemenin nerelerde kullanıldığına bakacak olursak;

- Demografik bilgi analizi.
- Yüz tanıma ve güvenlik sistemleri
- Uydu verilerinden yararlanılarak nüfus yoğunluğunun tespit edilmesi.
- Astronomi, trafik, radar, fotoğraf ve endüstrisi uygulamaları.
- Radyoloji (Tomografi, Ultrason vb.).
- Endüstri ve askeri alanlarında görüntülerin ve uydu görüntülerinin anlamlandırılması.
- Tıp alanı
- Tarım ve Hayvancılık

gibi birçok alanda kullanılmaktadır.

3.2. Görüntü İşleme Süreçleri

Temel olarak;

- Görüntünün analiz edilmesi ve işlenmesi
- İmajın aktarılması

– Analiz edilen verilerin raporunun alınması

olarak üç adımda incelenmektedir.

3.3. Görüntü İşleme Teknikleri

Görüntü işleme teknikleri sırasıyla; görüntü ön işleme, görüntü geliştirme, görüntü ayırma, özellik çıkartma ve görüntü sınıflandırılması aşamalarından oluşmaktadır.

A) Görüntülerin Ön İşleme ve Geliştirme Süreci

Uydular ve dijital kameralar vasıtasıyla elde edilen görüntüler, görüntüleri yakalarken aydınlatma koşulları ve görüntüleme sistemlerinin sınırlamalarından dolayı parlaklık ve kontrast hataları oluşturmaktadır. Piksellerin geometrisi ve parlaklık değerleri ile ilgili hatalar görüntü ön işleme sayesinde engellenmekte olup oluşan hatalar matematik modeller (ampirik ya da teorik) kullanılarak düzeltilmektedir.

Görselin görünümünün iyileştirilmesi görüntünün makine veya insan yorumu için daha uygun hale dönüştürülmesi görüntü geliştirme sürecinde sağlanmaktadır. Temel olarak bu süreç piksel parlaklık değerleri üzerindeki oynamalarla sağlanmakta ve örnek olarak renklendirme, kenar ve kontrast geliştirme, görüntülerin filtrelenmesi, büyütme ve keskinleştirme gibi uygulamalar verilebilir.

B) Görüntü Bölütleme Süreci

Bu süreç, görüntüyü kendini oluşturan bileşenlere ve nesnelere ayırma işlemi olarak tanımlanmaktadır. Bölütleme işleminin ne zaman durdurulacağı, uygulama sürecinde işlenen konulara göre belirlenmekte olup görüntülerde aranan nesne veya bileşen görüntüden elde edildiği zaman bölütleme işlemi durdurulmaktadır.

Bu süreçte yöntem olarak görüntü eşiği yöntemleri kullanılmakta ve bu yöntemler kullanılırken nesne pikselleri gri seviyesine, arka plan ise farklı piksellere göre ayarlanmakta ve genelde nesne pikselleri siyah ve arka plan rengi ise beyaz olmaktadır. Piksel farkı nedeniyle ikili görüntülerin gri ölçeğe göre değerlendirilmesi yapılarak görüntü ayrımı elde edilmektedir.

Görüntünün bölümlere ayrılmış değeri $S(x, y)$, pikselin gri seviyesi $g(x, y)$, koordinatlardaki eşik değerleri $T(x, y)$ şeklinde ifade edilmiştir.

$$S(x, y) = \begin{cases} 1, & g(x, y) > T(x, y) \\ 0, & g(x, y) \leq T(x, y) \end{cases} \quad [2]$$

C) Özellik Çıkartma Süreci

Bu süreç görüntü bölütlemenin bir üst versiyonu olarak değerlendirilmektedir. Nesneleri arka plandan ayırmanın yanı sıra asıl amacı nesnelerin şekli, boyutu ve kompozisyonu gibi özelliklere göre nesneyi tanımlamaktır.

D) Görüntü Sınıflandırma Süreci

En sık kullanılan bilgi çıkarma yöntemlerinden birisi sınıflandırma işlemidir ve en genel şekliyle piksel ya da piksel grubunun gri değerine bağlı olarak etiketlenmesi şeklinde ifade edilmektedir. Etiketleme işlemi sırasında bir nesnenin çoklu özellikleri kullanılmaktadır. Bu sebeple sınıflandırma işlemi kolaylaştıran unsur aynı nesneye ait birçok görüntünün olmasıdır. Uzaktan algılama teknolojisi kullanan araçlarda, bilgi çıkarım teknikleri genellikle görüntülerin spektral yansıma özelliklerini analiz etmektedir. Spektral analizler, genel olarak, denetimli ve denetimsiz multispektral sınıflandırma olarak iki farklı sistemle yürütülmektedir.

4. MATERYAL ve YÖNTEM

Bu bölüm içerisinde, tez çalışmasında kullanılan materyaller ve notasyonlar işlenmiştir. Nesne tanımlamada kullandığımız, derin öğrenme algoritmasının yapısı, Nvidia Hesaplamalı Birleşik Cihaz Mimarisi CUDA mimarisi, YOLO modelleri, DetectNet, kullanılan materyaller, veri seti hazırlık süreçleri ve modellerin eğitimleri ile ilgili bilgi verilmiştir.

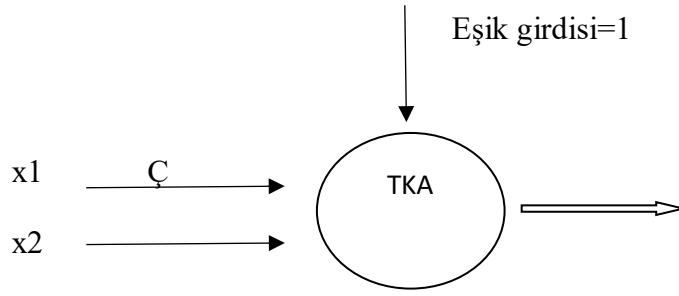
4.1. Derin Öğrenme

Genel bir ifadeyle, görüntülerin tanımlanması, bir bilgisayar konuşma tanıma veya tahmin yapma gibi insana ait benzer görevleri gerçekleştirilmesi için eğiten bir tür makine öğrenmesi algoritmaların bir araya geldiği kümeler topluluğu olarak tanımlanmaktadır. Derin öğrenme ile verilerin temel parametreleri oluşturulmakta ve bilgisayar, birçok proses katmanını kullanarak görüntüleri tanımak için eğitilmektedir. Derin öğrenme algoritmalarının yapısını, yapay sinir ağları kökenli modeller ve enerji tabanlı modeller oluşturmaktadır.

Yapay Sinir Ağı (YSA)'lar, birbirleriyle iletişim kurabilen ve birbirine bağlı işlem birimlerinden oluşan nöron adı verilen bir öğrenme sisteminden oluşmaktadır.

4.1.1. Tek Katmanlı Algılayıcılar

Sadece girdi ve çıktı katmanından oluşan Tek Katmanlı Algılayıcılar (Single Layer Perceptron – SLP), doğrusal problemlerin çözümünde kullanılmakta olup, katmanların bir ya da birden fazla nöronu bulunabilmektedir. Şekil 5.1.1'de SLP modeli yer almaktadır.



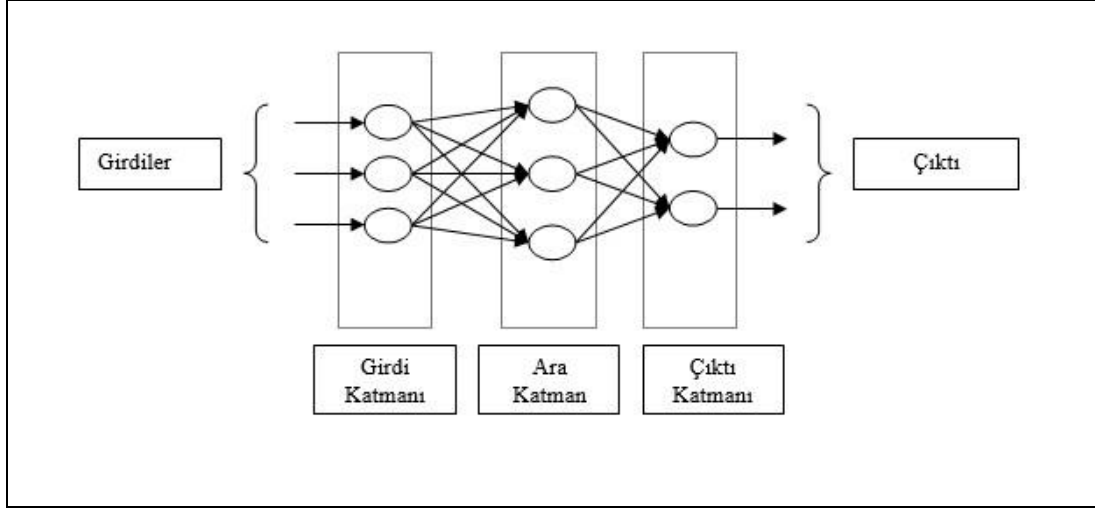
Şekil 3. Tek Katmanlı Algılayıcı (TKA) Modeli

Eşik değeri, bu tip ağlarda nöron elemanlarının değerlerinin ve de ağıın çıktısının 0 olmasını önler. Değeri daima 1'dir. Ağıın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu elde edilir. Tek katmanlı algılayıcılarda doğrusal bir fonksiyon olan çıktı fonksiyonları -1 veya 1 değerlerini almaktadır. Çıktı 1 ise birinci sınıfa, -1 ise ikinci sınıfa kabul edilmektedir.

4.1.2. Çok Katmanlı Algılayıcılar

Adaline yöntemleri ve Tek katmanlı algılayıcı doğrusal olmayan çözümler üretemediği için hem eğitim hem de mimari algoritması açısından iyileştirilmiş Çok Katmanlı Algılayıcı (Multi Layer Perceptron - MLP) ağı önerilmiş ve MLP, mimari açıdan doğrusal olmayan aktivasyon fonksiyonuna sahip birçok nöronun birbirine hiyerarşik olarak bağlandığı bir yapıya sahip olan, Algılayıcı ve Adaline yöntemlerinin avantajları yanı sıra geri-yayılım adındaki öğrenme sistemini kullanmaktadır. Genel olarak yapay sinir ağları, ileri beslemeli ve geri beslemeli ağlar olarak iki bölüme ayrılmaktadır.

MLP, bir özellik hiyerarşisi oluşturmak için mevcut yapısına ara katmanlar eklemektedir. MLP de sırasıyla girdi katmanı, gizli katman ve son olarak çıktı katmanı bulunmaktadır. Bütün bu katmanlar bütünüyle, bir katmanın çıkışı ve bir sonraki katmanın ise girişi olacak şekilde birbiri üzerine sıralanmaktadır. SLP' de olduğu gibi, bütün katmanlar tamamen birbirine bağlıdır. Birimler arasındaki bağlantılara ağırlık adı verilmektedir. Şekil 4'te, bir giriş katmanı, iki gizli katman ve çıkış katmanı olan bir MLP' nin mimarisi gösterilmektedir. MLP mimarilerinin hiper parametrelerini, ara katmanların sayısı ve ara katman başına birim sayısı oluşturmaktadır.



Şekil 4. Çok Katmanlı Algılayıcı

Afin Dönüşümü

Afin Dönüşümü (Affine Transformation), MLP'nin her katmanı tarafından gerçekleştirilmekte olup hesaplama mantığında, matris çarpımı ve ardından bir bias eklenmektedir (Denklemler 3).

$$\vec{a}^l = h^{l-1} \vec{W} + \vec{b}^l, \quad [3]$$

l : katman

h^{l-1} : l. Katmanın girdisi

W : Ağırlık Matrisi

b^l : Bias vektörü

a^l : Afin dönüşümünün çıktısı

Doğrusal Olmayan Sistemler

Aktivasyon fonksiyonunda bütün tabakaların ilgin dönüşümü çıktısına (a^l) uygulanmasıyla, doğrusal olmayanlık elde edilmektedir.

Hücreye gelen net girdi, aktivasyon fonksiyonu tarafından işlenmekte ve hücrelerde girdiye karşılık üreteceği çıktı belirlenmektedir. Genellikle doğrusal olmayan bir fonksiyon seçilmekte olup yapay sinir ağlarının “doğrusal olmama” özelliği aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Dikkat edilmesi gereken bir diğer nokta ise, aktivasyon fonksiyonunun türevinin kolay hesaplanabilir olmasıdır. Bu nedenle geri beslemeli ağlarda aktivasyon fonksiyonunun türevinin de kullanılması durumu göz önünde bulundurulduğunda, türevi kolay hesaplanır bir fonksiyon seçilmesi gerekir.

Tanjant Hiperbolik Fonksiyonu, en sık kullanılan aktivasyon fonksiyonlarından birisidir (Denklem 4). Burada “x” girdilerin ağırlıklandırılmış doğrusal birleşimini ifade etmekte, genellikle girdiler [0,1] aralığında olduğu durumda çalışmakta ve [-1,1] aralığında çıkış üretmektedir.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \quad [4]$$

Lojistik sigmoid fonksiyonu bir aktivasyon fonksiyonu olup yaygın olarak kullanılmaktadır (Denklem 5). Bu fonksiyonun sık kullanılma sebeplerinden biri, her noktada sigmoid işlevinin diferansiyellenebilir olmasından kaynaklanmaktadır.

$$f(x) = \frac{1}{1 + e^{-x}} \quad [5]$$

Ağın doğrusal olmama durumunu artırmak amacıyla, Doğrultulmuş Lineer Birim Fonksiyonu (Rectified Linear Unit's Function - ReLU) (Denklem 6) kullanılır. Gelen girdi değerinin 0’ dan büyük olmaması durumunda aynen, yani değişmeden çıktı olarak verilmektedir. Bir nevi doğrusal fonksiyon gibi çalışmaktadır. 0’ dan küçük eşit olması durumunda ise direkt 0 olarak çıkmaktadır. Daha basit ve diğer fonksiyonlara göre daha hızlıdır. Bu sebeple, DSA’ larda çok sık kullanılır.

$$f(x) = \max(0, x) \quad [6]$$

4.1.3. Eğitim

Kayıp Fonksiyonu

Kayıp fonksiyonu (maliyet-loss function), genellikle ağ sonucunun doğruluğunu ölçmek için kullanılmaktadır. Tahmin değerinin, beklenen değerden ne kadar farklı olduğunu ifade etmektedir. Çapraz-entropi (cross-entropy) kayıp fonksiyonu genellikle görsel sınıflandırma problemlerinde kullanılır (Denklem 7).

$$L = - \sum_{i=1}^m y_i \log_2 p_i \quad [7]$$

“y” hedef vektörü, “m” çıkış katmanlarındaki olası sınıfların sayısını, “ p_i ” ise tahmin edilen olasılık değerini ifade etmektedir.

Geri Yayılm

Bir sinir ağı eğitim algoritması olan geri yayılım algoritması, denetimli öğrenme sınıfında olduğu için hedef sınıflarda hata hesaplaması için gereklidir. “e” hatayı, “x” girdiyi, ifade etmekte olup “ ϕ ” ise aktivasyon fonksiyonu verilen her katmandaki ağırlığa (w_{kj}) göre kayıp fonksiyonunun (L) bir gradyanti olarak elde edilmektedir (Denklem 8).

$$a_j = \phi \left(\sum_{k=1}^n w_{kj} x_k \right) \quad [8]$$

$$e = \frac{\partial L}{\partial w_{kj}} = \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial w_{kj}} \quad [9]$$

Gradyan hesaplamasında, belirli ağırlık w_{kj} ' ye göre kayıp fonksiyonunun L 'nin kısmi türevi hesaplanmakta ve zincir kuralı uygulanmaktadır. “e” değeri kullanılarak ise ağırlıklar gradyanti inişi gibi bir optimizasyon algoritması tarafından güncellenmektedir.

Gradyan Düşüş

Sinir ağları içerisinde en yaygın kullanılan optimizasyon algoritmalarından birisidir ve yaygın olarak kullanılan ifadeyle, bir fonksiyon üzerinde rastgele bir noktadan başlanarak bu noktanın koordinatlarını, kısmi türevlerin tersi (gradyan vektörünün tersi) yönünde değiştirerek küçük adımlarla minimum noktaya yaklaşma yöntemine gradyan inişi denmektedir. Adımların büyüklüğünü ise adım boyutu adı verilen çarpan belirlemekte ve adım boyutuna öğrenme oranı da denmektedir olup bu oran $[0,1]$ bir arada bir değerdir. Ağırlıklarda güncelleme yapmak için, tüm eğitim seti kullanılmaktadır.

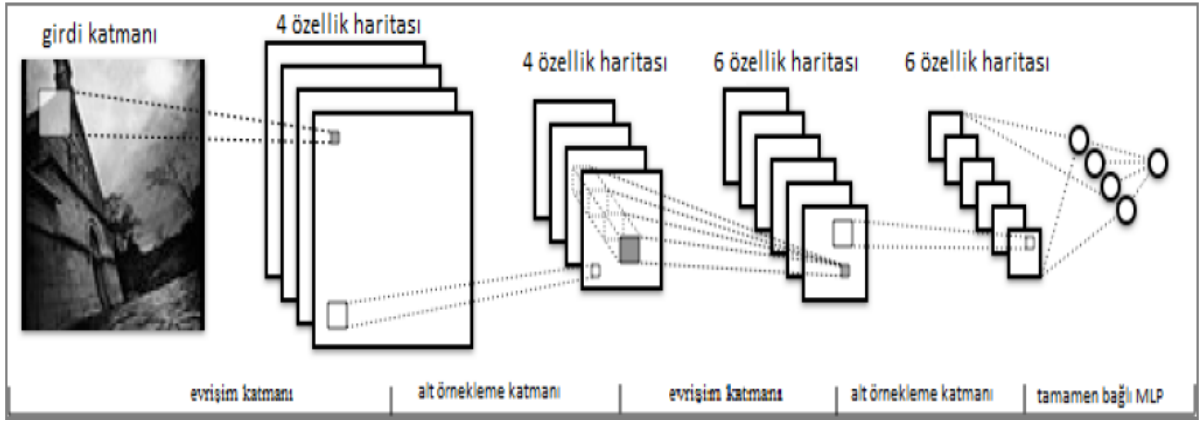
4.1.4. Evrişimsel Sinir Ağı

Evrişimsel sinir ağı (ConvNet / Convolutional neural networks -CNN), giriş değeri olarak bir görüntü değeri verildiğinde görüntülerdeki farklı görünüşleri ve nesneleri tek tek ayırabilen derin öğrenme algoritmasıdır.

Evrişimli sinir ağların temel görevleri arasında görüntülerin sınıflandırılması, benzerlik kullanarak kümeleme işlemi yapmak ve nesne tanıma yer almaktadır.

Evrişimsel Sinir Ağı' nın Temel Yapısı

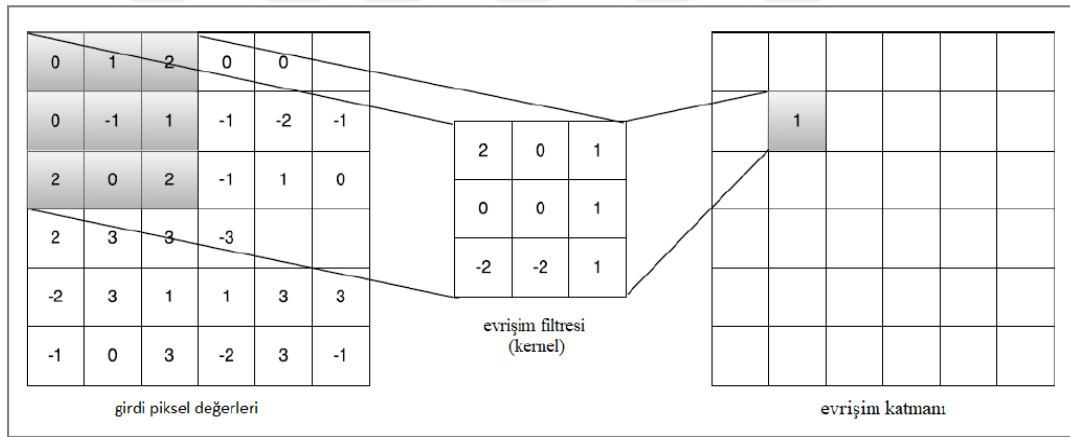
Temel olarak ağ, evrişim katmanlarından (convolution layers), alt örnekleme katmanlarından (pooling layers) ve bağlı katmanlardan (fully connected layers) oluşmaktadır. Ağın yapısı Şekil 5'teki gibidir.



Şekil 5. Görüntü İşleme Süreci. [12]

a) Evrişim Katmanı

Bu katmanda, 2D görüntü evrişimi gerçekleştirilmekte ve filtrelenmiş yeni bir görüntü elde edilmektedir. Şekil 6'da örnek bir 2D bir görüntünün evrişim işlemi gösterilmiştir.



Şekil 6. Görüntü İşleme Süreci. [12]

Evrişim işleminin her adımında, filtrede yer alan ögeler, alıcı alanda karşılığı olan piksel değerleri ile çarpıldıktan sonra birleşmiş değerleri elde etmek için toplanmaktadır. Filtreler, tüm girdi kümesinin üzerinde kaydırılmakta ve böylece birleştirilmiş aktivasyon haritaları oluşmaktadır. Şekil 6'da evrişim işlemine tabi tutulmuş örnek bir aktivasyon haritası gösterilmiştir. Evrişim işleminden sonra bias (b_i) terimi genellikle eklenmektedir.

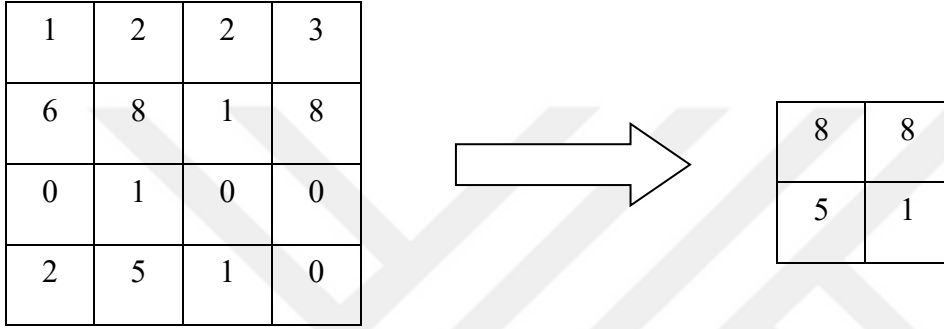
$$\overrightarrow{y_{ijm}} = (\overrightarrow{W_i} * \vec{X})_{jm} + \overrightarrow{b_i} \quad [10]$$

y_{ijm} i . çıktı özellik haritasının (j,m) koordinatının piksel değerini x , W_i i . evrişim filtresini, x girdi değerini belirtmekte, b_i i . evrişim filtresine karşılık gelen bias vektörün i . elemanını göstermektedir.

b) Alt Örnekleme Katmanı (Pooling Layer)

Hesaplama maliyetini düşürmek için Alt Örnekleme Katmanı, ağ genelinde evrişim işlemleri ile üretilen girdi verilerini alt örneklemeye yapar. Böylelikle girdilerin boyutları, ağıdaki parametrelerin sayıları ve hesaplama maliyetleri azaltılmaktadır.

Alt örneklemeye için farklı modeller önerilmiş olup en sık kullanılanlar şunlardır: maksimum alt örneklemeye (max pooling), ortalama alt örneklemeye (average pooling) ve rasgele alt örneklemeye (stochastic pooling). Şekil 7’de bir maksimum alt örneklemeye işlemi görülmektedir.



Şekil 7. Görüntü İşleme Süreci. [12]

c) Aktivasyon Katmanı

Özellik haritalarında, her bir öge için doğrusal olmayan bir aktivasyon fonksiyonu kullanılmaktadır. Bunun için sinir sistemleri örnek alınmış ve uyarıların nasıl aktarıldığı taklit edilmektedir. En sık kullanılanlar arasında hiperbolik tanjant fonksiyonu, sigmoid fonksiyonu ve doğrultucu lineer birim fonksiyonu (ReLU' lar) yer almaktadır.

d) Tamamen Bağlı Katmanlar

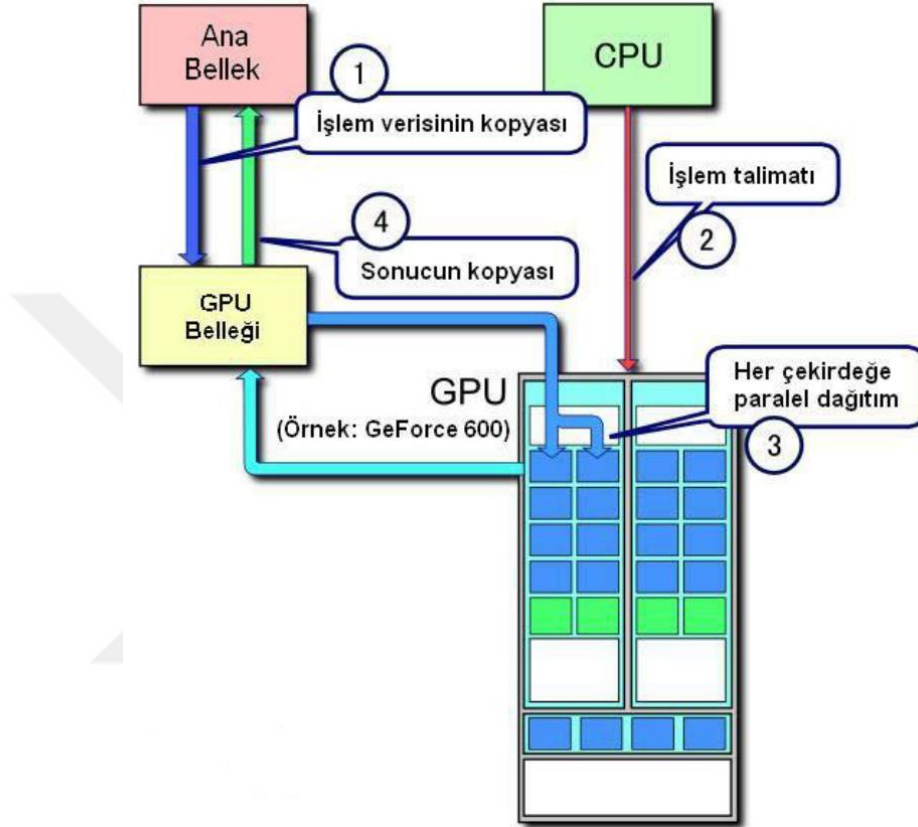
Katmandaki yapı modeli, geleneksel çok katmanlı algılayıcılar (MLP) yapısıyla ile aynı olup tamamen bağlı katman, her çıkış nöronunun bir önceki katmandaki tüm giriş nöronuna bağlı olması durumunu ifade etmektedir.

e) Sınıflandırıcı

Sınıflandırıcı, tamamen bağlantılı katmanların çıktılarını girdi olarak almakta ve sınıflar üzerindeki olasılık dağılımını hesaplamaktadır. Çok kategorili sınıflandırmalarda en çok kullanılan sınıflandırıcılardan biri de softmax sınıflandırıcılardır.

4.2. CUDA

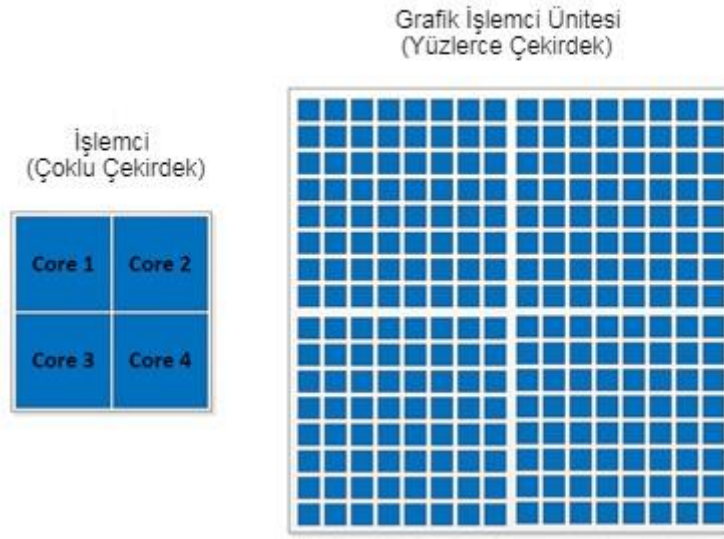
CUDA, NVIDIA tarafından geliştirilen ve 23 Haziran 2007'de tanıtılan GPU' lar için bir mimaridir. CUDA, “Compute Unified Device Architecture” adının kısaltılmış hali olarak yayınlanmış paralel işlemekten yararlanan bir yazılım mimarisidir. Aşağıda CUDA işlem akışı gösterilmiştir.



Şekil 8. CUDA İşlem Akışı

GPU mimarisi ve Geleneksel CPU donanım mimarisi Şekil 9'da gösterilmiş olup genel amaçlı grafik işlem birimi (GPGPU- General Purpose Graphics Processing Units) olarak ifade edilmektedir.

CPU'da az sayıda aritmetik lojik üniteleri (ALU) ve geniş kontrol ünitesinin beraberinde, paylaşımlı önbellek büyük yer kaplamaktadır. GPU' larda ise çok sayıda küçük kontrol üniteleri ve paylaşımlı önbellek yer almakta olup daha çok sayıda çekirdeğe bloklar halinde atanıp işleyişi yönetmektedirler.



Şekil 9. CPU ve GPU mimarilerinin kıyaslanması. [13]

CPU ve GPU Arasındaki Fark

GPU ve CPU mimarileri arasındaki farklar göz önünde bulundurulduğunda en önemli fark, CPU'ların çok çeşitli görevleri hızlı bir şekilde işlemek üzere tasarlanmış olmasıdır, fakat çalıştırılabilen görevlerin eşzamanlılığı sınırlıdır.

GPU'lar CPU'lar kadar çok yönlü değilken, GPU'lar, büyük paralellik nedeniyle verileri bir CPU'dan birkaç kat daha hızlı işleyebilmektedir. Aynı zamanda CPU'lar, GPU'nun yapamadığı bir bilgisayarın her giriş ve çıkışını yöneten büyük ve geniş talimat setlerine sahiptir. Bir sunucu ortamında 24 ila 48 çok hızlı CPU çekirdeği olabilir. Aynı sunucuya 4 ila 8 GPU eklemek, 40.000'e kadar ek çekirdek sağlayabilir.

GPU'lar, tekrarlayan ve oldukça paralel bilgi işlem görevleri için en uygun olanıdır. Video işlemenin ötesinde, GPU'lar finansal simülasyonlarda, makine öğreniminde, risk modellemede ve diğer birçok bilimsel hesaplama türünde oldukça iyi hizmet vermektedirler. Geçmiş yıllarda, GPU'lar Bitcoin veya Ethereum gibi kripto para birimleri madenciliği için kullanılırken, GPU'lar artık genellikle ölçekli olarak kullanılmamakta ve Alan Programlanabilir Izgara Dizileri (FPGA) ve ardından Uygulamaya Özel Entegre Devreler (ASIC) gibi özel donanımlara yol açmaktadır.

CPU'dan GPU'ya Hesaplama Örnekleri

CPU ve GPU oluşturma videosu — Grafik kartı, videoyu bir CPU'ya güvenmekten daha hızlı bir şekilde bir grafik biçiminden diğerine dönüştürmeye yardımcı olur.

Verileri hızlandırma — Bir GPU, belirli bir süre içinde bir CPU'nun işleyebileceği veri miktarını hızlandıran gelişmiş hesaplama yeteneğine sahiptir. Derin öğrenme veya makine öğrenimi gibi karmaşık matematiksel hesaplamalar gerektiren özel programlar olduğunda, bu hesaplamalar GPU tarafından yüklenebilir. Bu, CPU'nun diğer görevleri daha verimli bir şekilde tamamlaması için zaman ve kaynak tasarrufu sağlar.

Kripto para madenciliği — Bitcoin gibi sanal para birimleri elde etmek, işlemleri işlemek için bir bilgisayarın röle olarak kullanılmasını içerir. Bir CPU bu görevi yerine getirebilirken, grafik kartındaki bir GPU, bilgisayarın çok daha hızlı para birimi oluşturmaya yardımcı olabilir

4.3. YOLO

YOLO, 'You Only Look Once' teriminin kısaltmasıdır. Bu, bir resimdeki çeşitli nesneleri (gerçek zamanlı olarak) algılayan ve tanıyan bunun için sinir ağlarını kullanan bir algoritmadır. YOLO algoritmasında nesnelerin tespitleri regresyon problemindeki çözüm mantığı yapılmakta ve tespit edilen görüntülerin sınıf olasılıkları belirlenmektedir.

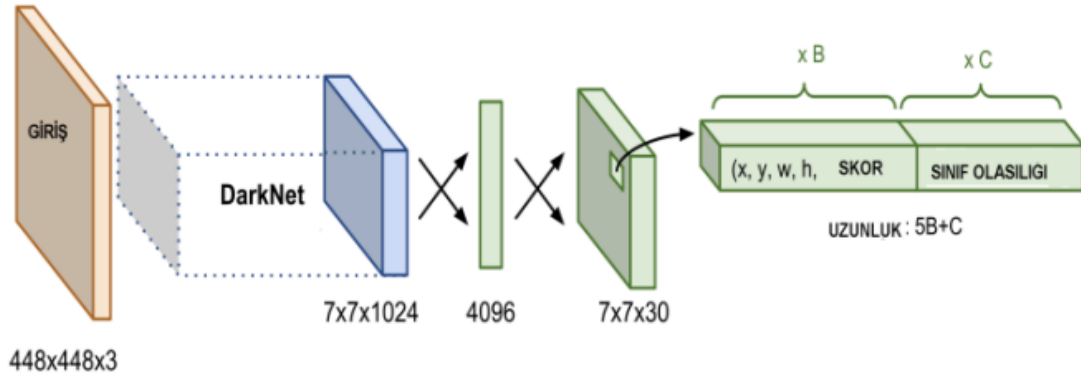
YOLO algoritması, nesneleri gerçek zamanlı olarak algılamak için evrişimli sinir ağlarını (CNN) kullanır. Adından da anlaşılacağı gibi, algoritma nesneleri algılamak için bir sinir ağı üzerinden yalnızca tek bir ileri yayılım gerektirir.

Bu, tüm görüntüdeki tahminin tek bir algoritma çalıştırmasında yapıldığı anlamına gelir. CNN, çeşitli sınıf olasılıklarını ve sınırlayıcı kutuları aynı anda tahmin etmek için kullanılır. YOLO algoritması çeşitli varyantlardan oluşur. Yaygın olanlardan bazıları küçük YOLO, YOLOv3, YOLOv4 ve YOLOv5'tir.

YOLO algoritması, ImageNet üzerinde eğitilmiş 53 katmanlı bir ağa sahip olan bir Darknet çeşidi kullanılmaktadır. Algılama işlemi için, üzerine 53 katman daha yığılmakta ve bu da bize YOLO için 106 kat tamamen evrişimli bir mimari sağlamaktadır.

	Tip	Filtre Boyutu	Çıktı
1x	Evrişimsel	32 3 × 3	256 × 256
	Evrişimsel	64 3 × 3 / 2	128 × 128
	Evrişimsel	32 1 × 1	
	Evrişimsel	64 3 × 3	
	Kalıntı		128 × 128
2x	Evrişimsel	128 3 × 3 / 2	64 × 64
	Evrişimsel	64 1 × 1	
	Evrişimsel	128 3 × 3	
	Kalıntı		64 × 64
	Evrişimsel	256 3 × 3 / 2	32 × 32
8x	Evrişimsel	128 1 × 1	
	Evrişimsel	256 3 × 3	
	Kalıntı		32 × 32
	Evrişimsel	512 3 × 3 / 2	16 × 16
	Evrişimsel	256 1 × 1	
8x	Evrişimsel	512 3 × 3	
	Kalıntı		16 × 16
	Evrişimsel	1024 3 × 3 / 2	8 × 8
	Evrişimsel	512 1 × 1	
	Evrişimsel	1024 3 × 3	
4x	Kalıntı		8 × 8
	OrtHavuz		
	Bağlı Softmax		
	Küresel	1000	

Şekil 10. Darknet-53 Mimarisi. [15]



Şekil 11. Darknet Görüntü İşleme Süreci. [15]

Darknet-53'e ve ResNet'e göre, daha iyi performans göstermektedir ve 1,5 kat daha hızlıdır. Kısa yol bağlantıları içeren Darknet-53 ve özellik birleştirme ve harita örnekleme özelliklerine sahiptir, ayrıca kayan nokta işlemlerini başarılı bir şekilde gerçekleştirmektedir. Bu durum Darknet-53'ün ağ yapısının GPU'yu daha verimli ve iyi bir şekilde kullandığını göstermektedir. YOLO kamyon, araba, yaya, bisikletli ve hatta hayvanları tanımlama yetisine ve son derece hızlı bir algoritmaya sahiptir. YOLO da görüntüler 416x416 pikselde 22 milisaniyede çalışmaktadır ve SSD (Single Shot Detection) kadar hassas ve 3 kat daha hızlıdır. YOLO diğer ağ yapılarına göre çok daha iyi bir ağ yapısına sahip olmakla birlikte diğer sistemlere göre yaklaşık 4 kat daha hızlıdır. YOLO'nun en önemli özelliği ise yeniden eğitime ihtiyaç

duymamasıdır. Diğer CNN tabanlı sistemlerin çoğu görüntüdeki bir nesneyi algılamak için sınıflandırıcıları yeniden kullanır. Yaygın olan nesne algılama yöntemleri ile kıyaslandığında, nesnenin sınıflandırılması ve konumlandırılması için tek CNN uygulanmaktadır [14].

Bu algoritma, hızı ve doğruluğu nedeniyle popülerdir. Trafik sinyallerini, insanları, parkmetreleri ve hayvanları algılamak için çeşitli uygulamalarda kullanılmıştır.

YOLO algoritması aşağıdaki nedenlerden dolayı önemlidir:

Hız: Bu algoritma, nesneleri gerçek zamanlı olarak tahmin edebildiği için algılama hızını artırır.

Yüksek doğruluk: YOLO, minimum arka plan hatasıyla doğru sonuçlar sağlayan tahmine dayalı bir tekniktir.

Öğrenme yetenekleri: Algoritmanın, nesnelerin temsillerini öğrenmesini ve bunları nesne algılamada uygulamasını sağlayan mükemmel öğrenme yetenekleri vardır.

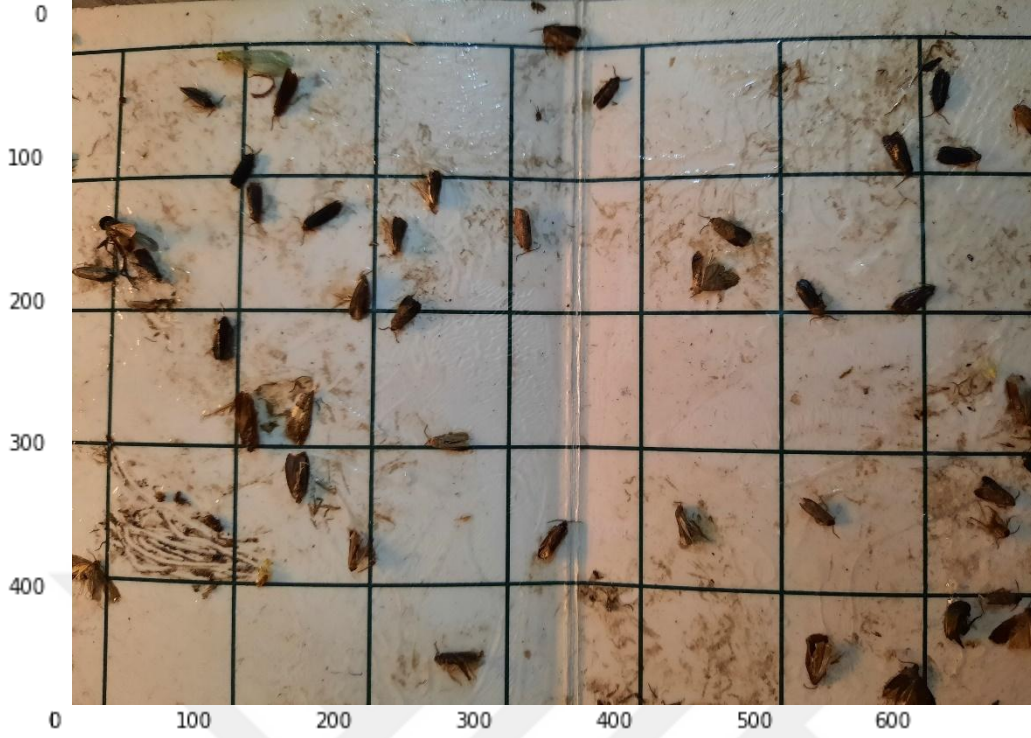
YOLO algoritmasının Çalışma Prensipleri:

YOLO algoritması aşağıdaki üç tekniği kullanarak çalışır:

- Artık bloklar
- Sınırlayıcı kutu regresyonu
- Birleşim üzerinden kesişme (IOU)

Artık Bloklar

İlk olarak, görüntü çeşitli ızgaralara bölünmektedir. Her ızgara $S \times S$ boyutuna sahiptir. Aşağıdaki Şekil 12’de yer alan görüntü, bir girdi görüntüsünün ızgaralara nasıl bölündüğünü gösterir.



Şekil 12. Nesnelerin Izgara Hücrelerinde Gösterimi

Yukarıdaki resimde, eşit boyutta birçok ızgara hücresi vardır. Her ızgara hücresi, içinde görünen doğu meyve güvesi nesnelerini algılayacaktır. Örneğin, belirli bir ızgara hücresi içinde bir nesne merkezi belirirse, bu hücre onu algılamaktan sorumlu olacaktır.

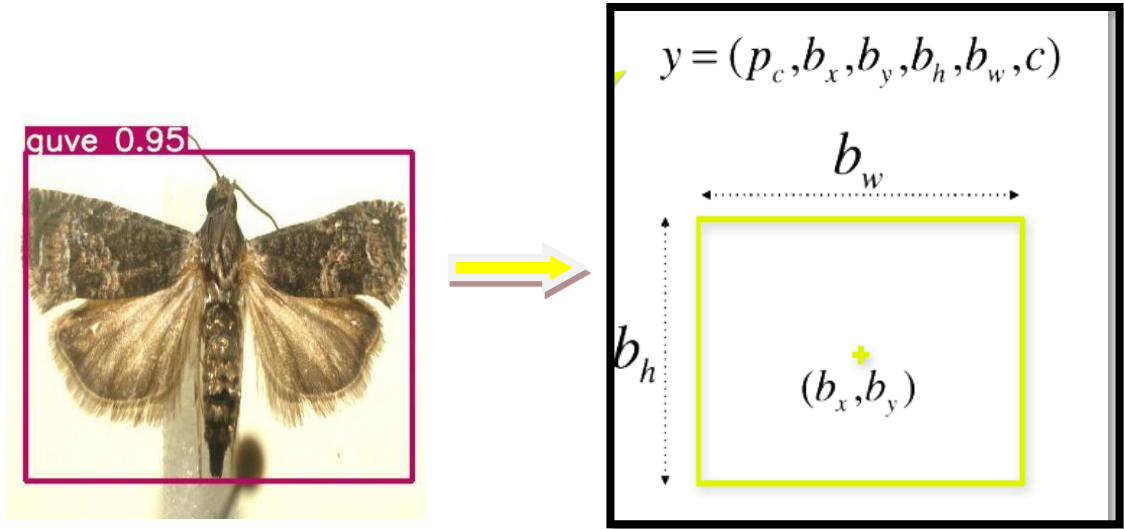
Sınırlayıcı Kutu Regresyonu

Sınırlayıcı kutu, görüntüdeki bir nesneyi vurgulayan bir ana hattır.

Görüntüdeki her sınırlayıcı kutu aşağıdaki özniteliklerden oluşur:

- Genişlik (bw)
- Yükseklik (bh)
- Sınıf (örneğin, kişi, araba, trafik ışığı vb.)- Bu, c harfi ile temsil edilir.
- Sınırlayıcı kutu merkezi (bx, by)

Aşağıda Şekil 13 de yer alan resimde bir sınırlayıcı kutu örneği gösterilmektedir. Sınırlayıcı kutu sarı bir çerçeve ile temsil edilmiştir.



Şekil 13. Sınırlayıcı Kutu Örneği

YOLO, nesnelerin yüksekliğini, genişliğini, merkezini ve sınıfını tahmin etmek için tek bir sınırlayıcı kutu regresyonu kullanır. Yukarıdaki resimde, sınırlayıcı kutuda görünen bir nesnenin olasılığını temsil eder.

- **Bağlı Sınıf Olasılığı:** Modelimizde kaç farklı sınıf varsa o kadar sayıda tahmin değeri anlamına gelmektedir. Örneğin Şekil 13’ de güve nesnesinin orta noktasının yakalandığı ızgarada güve nesnesinden eminse 1, farklı bir nesne olsaydı 0,2 gibi değerler atacaktı.

Birleşim Üzerinden Kesişme

Birleşim üzerinden kesişme (IOU), kutuların nasıl üst üste bindiğini açıklayan nesne algılamada bir olgudur. Intersection Over Union (IOU); dikdörtgenlerin kesiştiği bölgenin alanının, dikdörtgenlerin birleşim alanına bölünmesiyle hesaplanır. YOLO algoritması, nesneleri en iyi şekilde çevreleyen bir çıktı kutusu sağlamak için IOU'yu kullanmaktadır.

Bütün ızgara hücrelerinde, sınırlayıcı kutuları ve bunların güven puanlarını tahmin etmekten sorumludur. Öngörülen sınırlayıcı kutu gerçek kutuyla aynıysa, IOU 1'e eşittir. Bu mekanizma, gerçek kutuya eşit olmayan sınırlayıcı kutuları ortadan kaldırır.

Aşağıdaki görüntü, IOU' nun nasıl çalıştığına dair basit bir örnek sağlar.

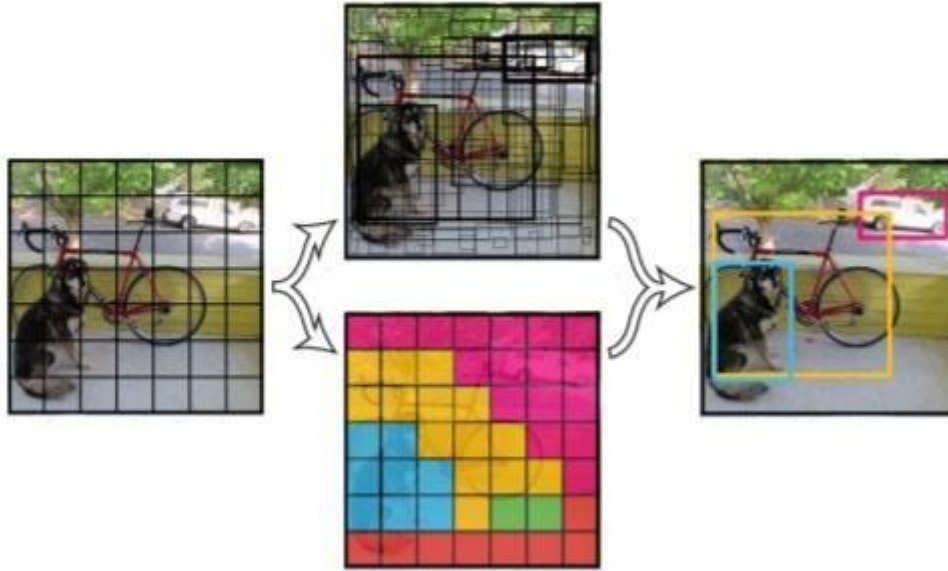


Şekil 14. Tahmin edilen Sınırlayıcı Kutu ve Gerçek Sınırlayıcı Kutu

Yukarıda yer alan Şekil 14’teki resimde, biri yeşil, diğeri mavi olan iki sınırlayıcı kutu vardır. Mavi kutu gerçek kutu iken yeşil kutu tahmin edilen kutudur. YOLO, iki sınırlayıcı kutunun eşit olmasını sağlar.

Üç tekniğin kombinasyonu

Aşağıdaki görüntü, nihai tespit sonuçlarını üretmek için üç tekniğin nasıl uygulandığını göstermektedir.



Şekil 15. Üçlü Teknik Gösterimi. [20]

Öncelikle görüntü, ızgara hücrelerine bölünmektedir. Her ızgara hücresi B sınırlayıcı kutuları tahmin etmekte ve güven puanlarını sağlamaktadır. Hücreler, her nesnenin sınıfını oluşturmak

için sınıf olasılıklarını tahmin etmektedir.

Örneğin, Şekil 15'e bakıldığında en az üç nesne sınıfı görebiliriz: araba, köpek ve bisiklet. Tüm tahminler, tek bir evrişimsel sinir ağı kullanılarak aynı anda yapılmaktadır.

Birleşim üzerinden kesişme, öngörülen sınırlayıcı kutuların nesnelerin gerçek kutularına eşit olmasını sağlar. Bu fenomen, nesnelerin özelliklerini (yükseklik ve genişlik gibi) karşılamayan gereksiz sınırlayıcı kutuları ortadan kaldırmaktadır. Nihai algılama, nesnelere mükemmel şekilde uyan benzersiz sınırlayıcı kutulardan oluşacaktır.

Örneğin, araba pembe sınırlayıcı kutuyla, bisiklet ise sarı sınırlayıcı kutuyla çevrilidir. Köpek, mavi sınırlayıcı kutu kullanılarak vurgulanmıştır.

Güven skoru hesabı

Güven Skoru = Kutunun Güven Skoru* Bağlı Sınıfın Olasılığı

Kutu Güven Skoru = $P(\text{nesne}) * \text{IoU}$

Pnesne: Kutunun nesneyi kaplayıp kaplamaması durumunun olasılığı. (Nesne var mı yok mu durumu)

IoU: Nesneye ait gerçek kutu ile tahmin edilen kutu arasındaki değeri ifade eder.

YOLO Algoritmasının gelişim sürecine bakıldığında;

Yolo 1' de bir ızgarada birden fazla nesnenin olması durumu sorun olurken, Yolo 2' de bu durum algoritmaya monte edilen Anchor Boxes methodu ile ilk olarak Faster R-CNN'de kullanılmış ve mantığına bakıldığında nesnenin etrafındaki kutuları tahmin etmemize olanak sağlamakla birlikte her bir ızgaradan önce belirlenen anchor box sayısı kadar tahmin yapılmasını sağlamamaktadır.

4.4. Materyaller

4.1.1. Bilgisayar

DSA'nın eğitilmesi için tez projesinde kullanılmış olan bilgisayara ait özellikler Tablo 1 de belirtilmiştir.

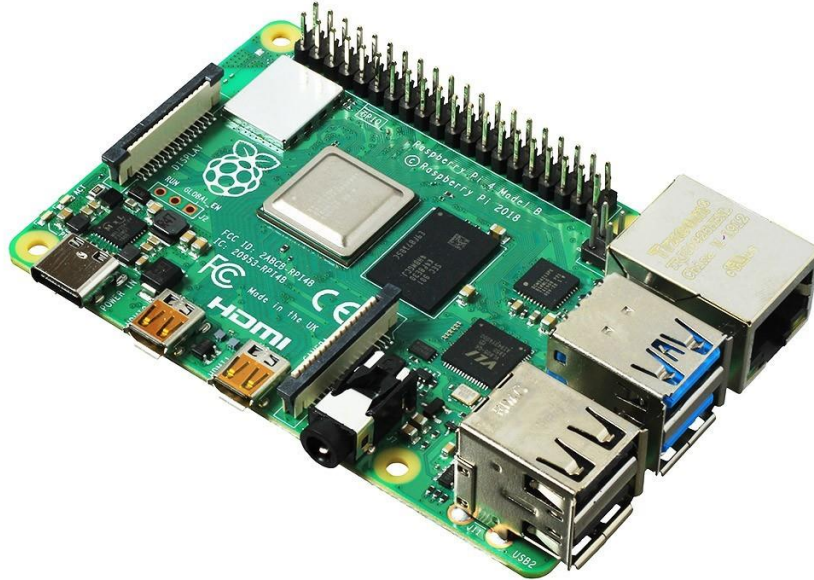
Tablo 1. Bilgisayar Özellikleri

İşlemci:	:	Intel(R) Core i7-6700HQ
İşlemci hızı:	:	Intel(R) Core i7-6700HQ up to 3.5 GHz
Ön bellek:	:	6 MB
Bellek:	:	12GB (1x12GB) DDR4L 1.2V 2133MHz
Disk kapasitesi ve arabirimi:	:	Samsung SSD 750 Evo 250GB ve 1TB Hitachi Travelstar
Ekran kartı:	:	4GB NVIDIA GeForce GTX1060 950M
Ekran:	:	15.6" Full HD 1920x1080 G-SYNC IPS Mat LED Ekran
İşletim sistemi	:	Windows 11 Home Insider Preview Single Language
Kullanım tipi	:	Dizüstü Bilgisayar
Üretici adı ve modeli	:	Asus X550V

4.1.2. Gömülü Kart

Birleşik Krallık ‘ta, Raspberry Pi Vakfı tarafından gelişmekte olan bütün ülkelerde en basit bilgisayar bilimi öğretilmesi amacıyla geliştirilen kredi kartı büyüklüğünde mini bilgisayardır [17]. Pi Zero ile 2012’de çıkan Raspberry Pi Model B zamanla geliştirilerek Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3 ve Raspberry Pi 4 modelleri oluşturulmuştur. Raspberry Pi, başta çocukların bilgisayarların çalışma mantığını anlaması ve basit kodlama yeteneği kazanması için geliştirilse de zamanla, internet tarayıcısı konusunda kısıtlı olsa da masaüstü bilgisayar yerine kullanma konusunda Pi mümkün hale getirildi. Daha bir sürü alanda kullanılmaktadır. Medya merkezi, hava durumu istasyonu, robot beyin, akıllı ev gereci, eski, drone yardımcısı, oyun konsolu ve daha birçok farklı cihaz üretmek amacıyla Pi kullanılmış ve yeni alanlarda kullanılmaya da devam edilmektedir [18].

Tez çalışmasında Raspberry Pi 4B 8GB modeli kullanılmış ve içerisine Raspbian İşletim Sisteminin kurulumu sağlanmıştır. Tablo 2’de Raspberry Pi 4B teknik özellikleri gösterilmiştir.

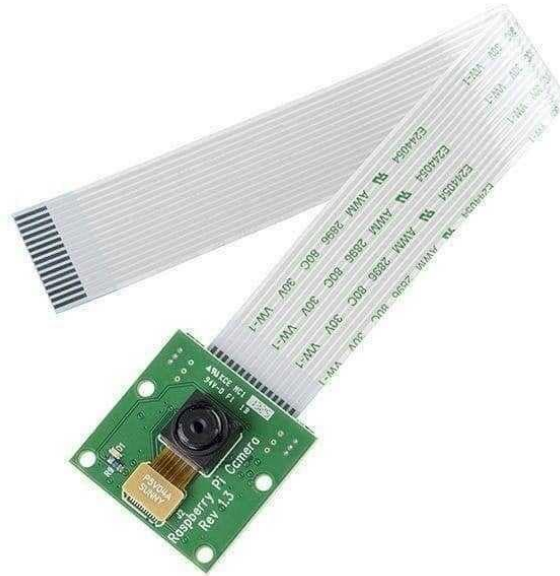


Şekil 16. Raspberry Pi 4. [19]

Tablo 2. Gömülü Kart Teknik Özellikleri

	Raspberry Pi 4B
İşlemci	Broadcom 2711 Dört çekirdekli Cortex-A72 64-bit destekli @1.5GHz
RAM	2GB/4GB LPDDR4-3200
Bağlantı	2.4GHz/5.0GHz 802.11.b/g/n/ac destekli kablosuz ağ Bluetooth 5.0, BLE destekli 2x USB 2.0 + 2x USB 3.0 portu Gerçek Gigabit Ethernet
Video ve Ses	2 x mikro-HDMI, 4K 60FPS destekli 1 X MIPI DSI ekran portu 1 X MIPI CSI kamera portu 4 kutuplu 3.5mm bağlantı üzerinden ses + kompozit video çıkışı
Multimedya	H.265 decode (4kp60) H.264 decode (1080p60) H.264 encode (1080p30) OpenGL ES 1.1, 2.0, 3.0 destekli grafik işlemci
Güç	USB-C tipi adaptör girişi 5V/3A PoE desteği (ek shield ile)

Tuzak içerisinde yapışkan levhaya yapışacak olan doğu meyve güvelerinin resimlerinin çekilmesi için Raspberry Pi 4B de kullanmak üzere uyumlu kamera modülünün teknik özellikleri Tablo 3’de verilmiştir.



Şekil 17. Raspberry Pi 4 Uyumlu Kamera Modülü

Tablo 3. Kamera Modülünün Teknik Özellikleri

Üretici Firma	China
Ürün Kategorisi	Raspberry Pi Kamera
Boyutları	25x20x9mm
Çözünürlük	5 MP (2592x1944 piksel)
Video Çekimi	1080p, 720p ve 640x480p
Sensör Tipi	OmniVision OV5647 (5 megapiksel)
Sensör Boyutu	3.67 x 2.74 mmI
Piksel Sayısı	2592 x 1944
Piksel Boyutu	1.4 x 1.4 um
Lens	f = 3,6 mm, f / 2,9
Görüş Açısı	54 x 41 derece
Görüş Alanı	2 m'de 2.0 x 1.33 m
Tam kare SLR lens eşdeğeri	35 mm
Sabit Odak	Sonsuza 1 m
Video	Codec H.264 (AVC) özellikli 30 fps' de 1080p
Pano boyutu	25 x 24 x 9mm (esnek kablo dahil değil)

4.1.3. GPRS Modülü

Raspberry Pi RGS-01 GSM Shield, SIM800, anten, Push-Push sim kart yuvası ile Raspberry PI uyumlu pinleri ile GSM şebekesi ile bağlantı kurmamızı sağlamakta olup akıllı tarım uygulamasında bahçe sahibine SMS gönderimini gerçekleştirecektir.

Ürün IMEI numarası bulunmakta ve Türkiye’de kayıtlıdır. Takılacak SIM kart ile internete bağlanma, SMS atma & alma, veri ve ses transferi yapmaya hazır bir Raspberry PI modülüdür.



Şekil 18. Raspberry Pi GSM Shield

4.1.4. Sıcaklık ve Nem Sensörü

DHT22 modelinde, sıcaklık ve nem algılayıcı kalibre edilmiş olup Raspberry Pi 4 e doğrudan ek bir bağlantı kablosuna gerek kalmadan, üzerinde yer alan pinler vasıtasıyla montajı yapılabilmektedir. Sıcaklık verilerimiz bu sensör aracılığıyla alınarak veri tabanımıza kaydedilmektedir.

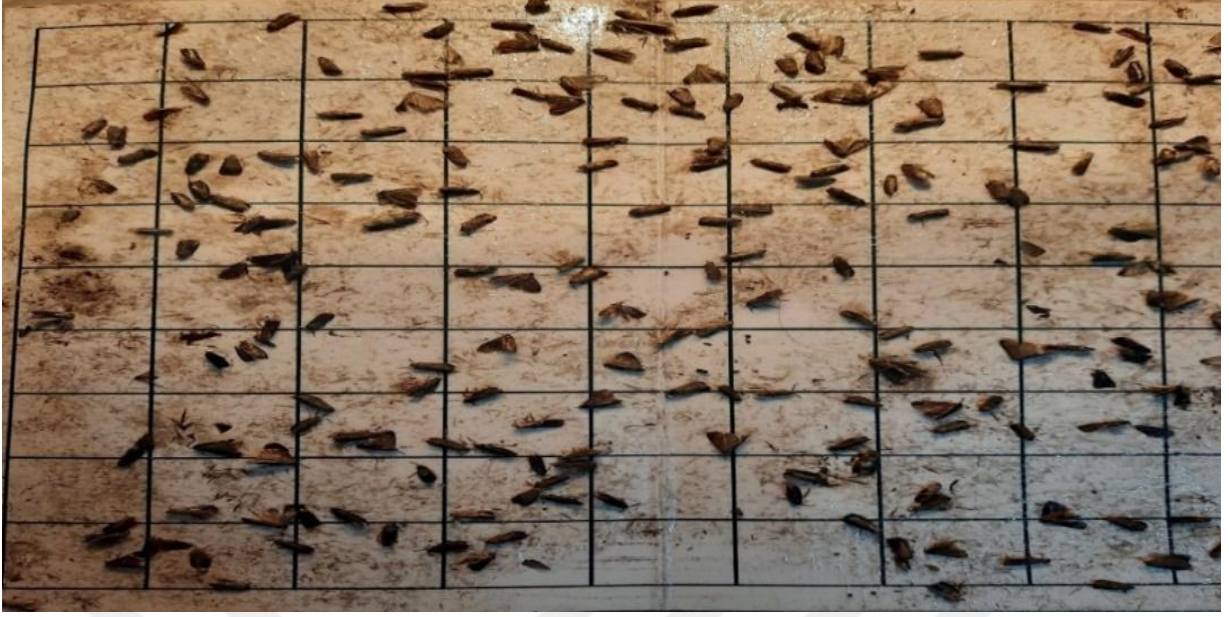
Tablo 4. DHT22 Sıcaklık ve Nem Sensörü Teknik Özellikleri

Boyutlar	28.2mm (Uzunluk) * 13.1mm (genişlik) * 10mm(yükseklik)
Ağırlık	6gr
Çalışma voltajı	3V-5.5V
Sensör Modelleri	AM2302 sıcaklık ve nemlilik sensörü
Sinyal Çıkışı	Dijital
Sıcaklık ölçüm aralığı	-40°C - 80°C
Ölçüm doğruluğu	0.5°C
Bağıl nem ölçüm aralığı	%0-100 RH
Bağıl nem ölçüm doğruluğu	%2.0 RH
Çözünürlük	16 bit

4.2. Veri Setinin Hazırlanması

4.2.1. YOLO Veri Setinin Hazırlanması

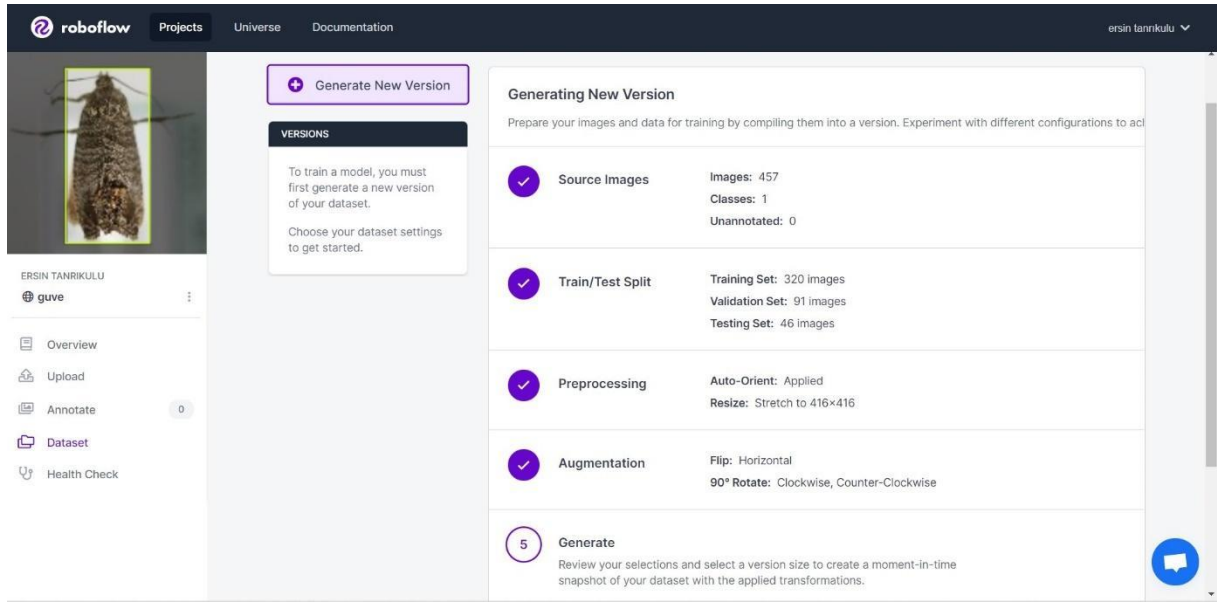
2021 yılı Nisan ayı içerisinde Yalova İlimizde deneme bahçelerine feromon tuzaklar kurulmuş ve tuzaklara içerisinde yer alan yapışkan levhalara doğu meyve güvesi dışısının kokusunu içeren kapsüller bırakılmıştır. Zamanla dişi kokusuna gelen yetişkinler yapışkan levha içeren tuzak içerisinde sayılarını artırmışlardır. Dolan levhalar tez çalışmamızda kullanılmak üzere muhafaza edilmiş ve levhalardaki doğu meyve güvelerinin fotoğrafları çekilmiştir.



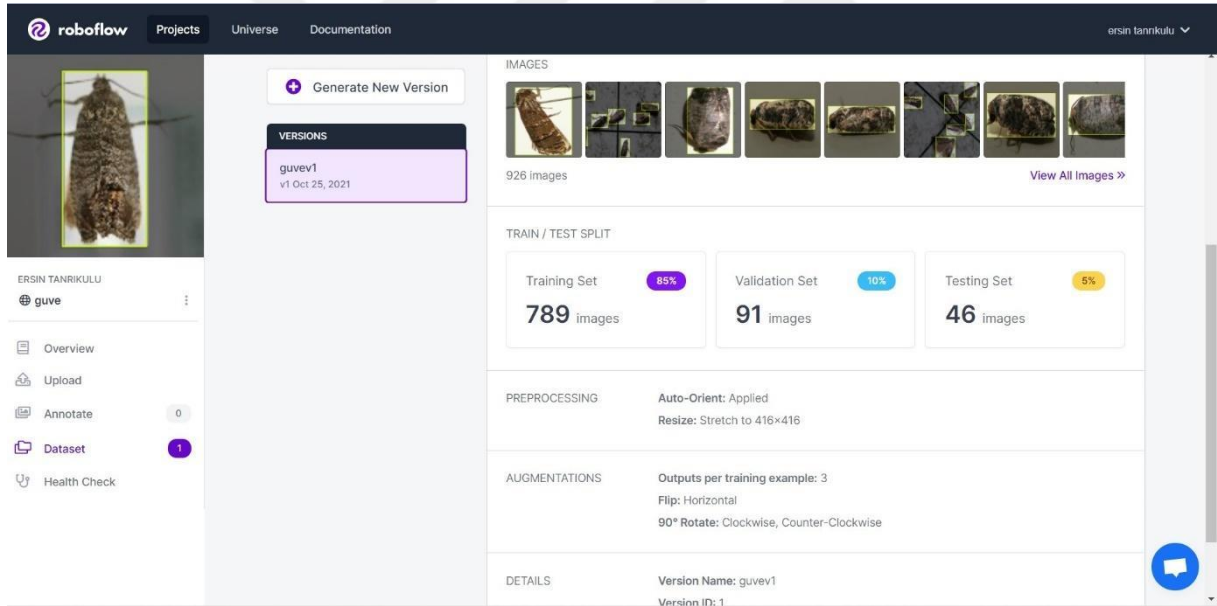
Şekil 19. Doğu Meyve Güveleri 2

Veri Artırma

Veri setlerinin büyüklüğü, derin öğrenme ve sınıflandırma modellerini etkilemektedir. Sıfırdan doğu meyve güvesi zararlısının veri setini oluşturmak zor ve zaman gerektiren bir problemdir. Çalışmamızın amacı, çektiğimiz mevcut görüntülerden roboflow ile yeni görüntüler üretip, bu görüntülerin doğu meyve güvesi tanıma sistemindeki etkisini analiz etmektir. Roboflow ile görüntüler üzerinde: yansıma, dönme, ölçekleme ve kırpma işlemleri ile görüntülerin miktarını artırarak derin sinir ağlarını eğitmek amacıyla kullanım hedeflenmiş olup çekilen 457 adet fotoğrafın 320 adeti training set, 91 adeti validation set ve 46 adeti ise testing set olarak ayrılmış, yatayda çevirme, 90 derece saat yönünde ve saat yönünün tersinde çoğaltma yöntemleri kullanarak veri seti 1097 adet etiketli fotoğrafa dönüştürülerek, Yolo v5 PyTorch formatında zip içinde jpg ve txt olarak dışarı aktarılmıştır.



Şekil 20. Roboflow İşlemleri



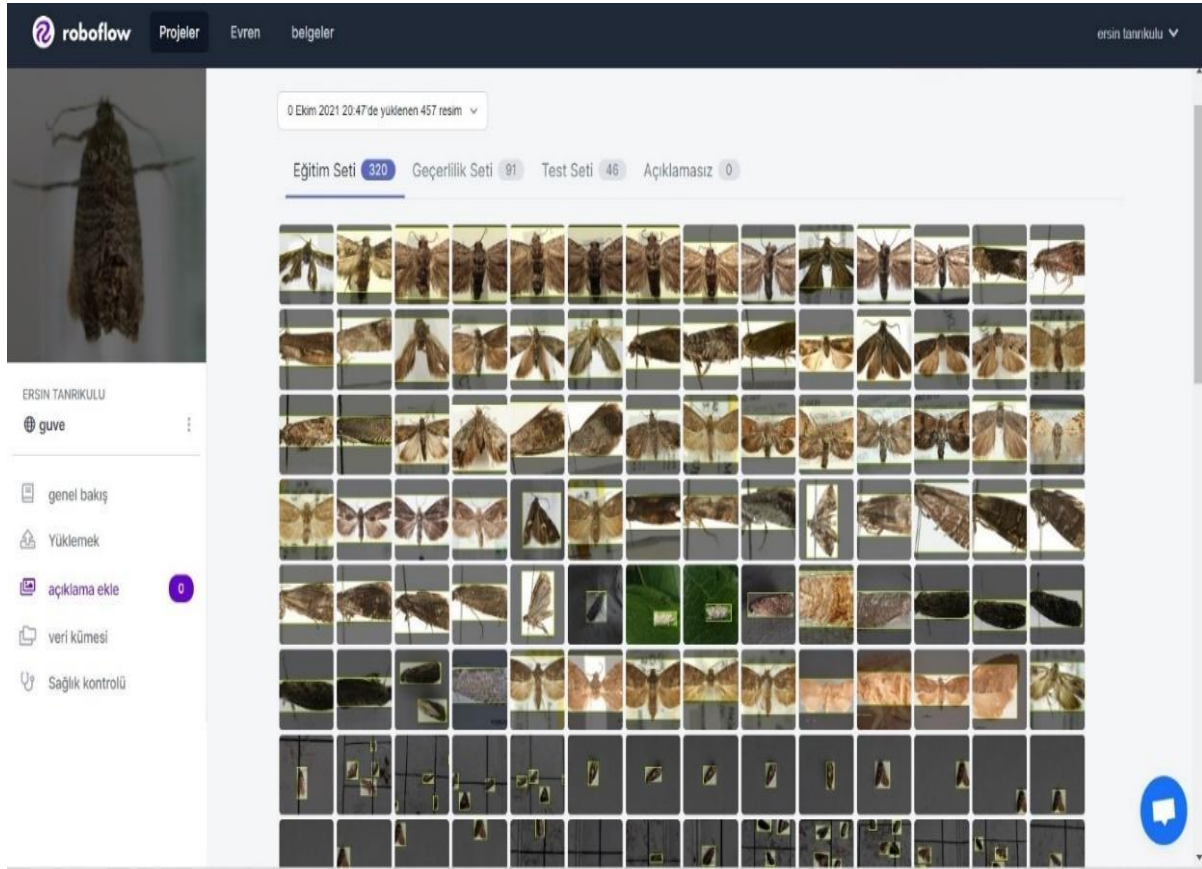
Şekil 21. Roboflow İşlemleri 2

Veri Temizleme

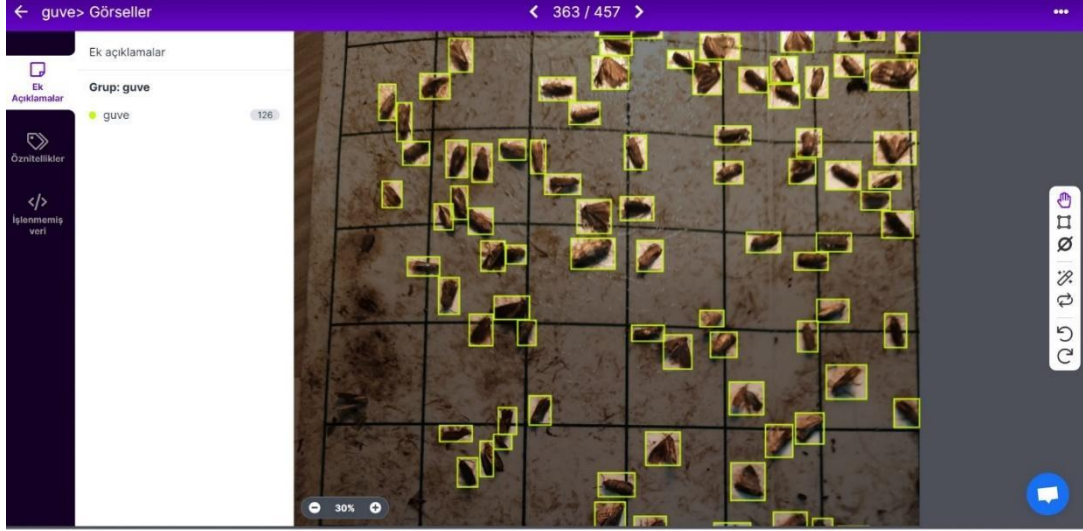
Veri artırma işlemi yapıldıktan sonra elde edilen veriler kontrol edilmiş ve kendini tekrar eden, benzer fotoğraf verileri veri setine dâhil edilmemiştir. Yapay sinir ağı modelinin eğitimi sırasında, kendini tekrar eden verilerin olması verilen veri setini ezberlenmesine (overfitting) neden olmaktadır. Bunu engellemek için veri seti kontrol edilerek benzeyen görüntüler veri setine dâhil edilmemiştir.

Veri Etiketleme

Verileri, Danışmanlı Öğrenme (Supervised Learning) yöntemi ile yapay sinir ağlarında kullanmadan önce etiketlemek gerekmektedir. Veri setinde bulunan görsellerin güve bulunan bölümleri etiketlenerek etiketli veri seti oluşturulmuştur. Küçük nesneleri doğru şekilde tespit etmemizi kolaylaştıran, verilerimizi içe veya dışa aktarırken düzinelerce format bulunmaktadır. Çalışma kapsamında çeşitli formatlar denemiş olup bunların arasından seçim yapmamızı sağlayan ve Colab Not Defterinde etiketlenmiş fotoğraflarımızla kolaylıkça işlem yapmamıza olanak sağlayan Roboflow programı kullanılmıştır.



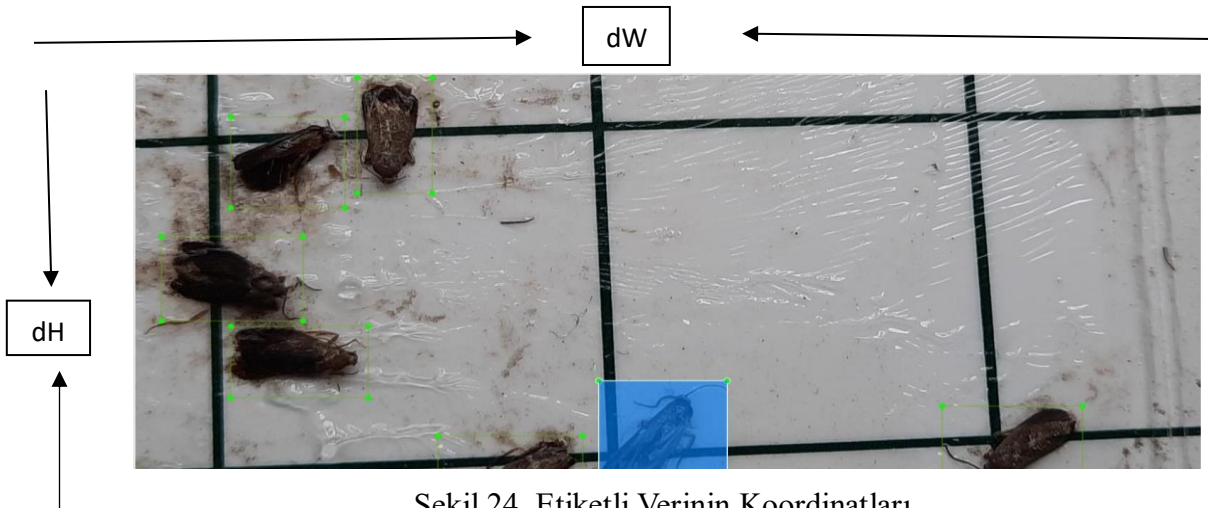
Şekil 22. Etiketli Güve Resimleri



Şekil 23. Etiketli Güve Resimleri

Veri Setinin Normalize Edilmesi

Etiketlenmiş görüntülerin genişliğini (dW) ve uzunluğunu (dH) olarak temsil edilmiştir. Güve nesnesinin konum bilgisini veren (X_0, Y_0) , (X_0, Y_1) , (X_1, Y_0) , (X_1, Y_1) noktaları ve görüntü üzerinde kaç adet güve olduğu bilgisi roboflow programı aracılığıyla text dosyası içerisine kaydedilmiş ve etiketli veri setini, sistemde kullanılan YOLOv5s mimarisine uygun hale getirmek için normalizasyon işlemi yapılmıştır. Daha sonra bu veriler Normalizasyon işlemi ile fotoğrafların (X_0, Y_0) , (X_0, Y_1) , (X_1, Y_0) , (X_1, Y_1) noktaları 0 ve 1 aralığına indirgenmiştir. Bu işlem ile görüntü içerisindeki etiketli güvenin, merkez noktası koordinatları (X, Y) , yüksekliği (H) ve genişliği (W) bilgileri elde edilmektedir. Yapılan normalizasyon işleminin formülü Denklem 11’de verilmiştir.



Şekil 24. Etiketli Verinin Koordinatları.

$A(X_0, Y_0)$ $B(X_1, Y_1)$

H

 $C(X_0, Y_1)$

W

 $D(X_1, Y_1)$

$$X = \frac{X_1 + X_0}{2} \times \frac{1}{dW}$$

$$Y = \frac{Y_1 + Y_0}{2} \times \frac{1}{dH}$$

$$W = (X_1 - X_0) \times \frac{1}{dW}$$

$$H = (Y_1 - Y_0) \times \frac{1}{dH}$$

[11]

X: Etiketlenmiş verinin orta noktasına ait X koordinatı.

Y: Etiketlenmiş verinin orta noktasına ait Y koordinatı.

W: Etiketlenmiş verinin genişliği

H: Etiketlenmiş verinin yüksekliği

Test ve Eğitim Sınıflarının Belirlenmesi

Denklem 11'de yer alan formüllere göre hesaplanan yeni koordinat, yükseklik ve genişlik değerleri (X, Y, W, H) görüntünün adı ile aynı olan yeni bir text dosyasına kaydedilmiştir. Ön işlemlerden sonra 1097 adet fotoğrafın %85'i öğrenme seti 789 adet, %10'u 91 adet doğrulama seti ve %5'46 adet test seti olmak üzere rastgele bir şekilde üçe ayrılmıştır.

4.3. Model Oluşturma ve Eğitim

Google Colab'ın sitesinde yeni bir Python Notebook'u oluşturulmuş ve sonrasında Notebook Settings kısmına gelip donanımı GPU seçilerek eğitim aşamasına geçilmiştir.

Roboflow-Train-YOLOv5 adlı not defterinin kopyası

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Kaydediliyor...

Dosyalar

- sample_data
- test
- train
- valid
- yolov5
 - README.dataset.txt
 - README.roboflow.txt
 - data.yaml

Epoch	gpu_mem	box	obj	cls	total	targets	img_size
297/949	1.86G	0.03047	0.03406	0	0.06453	21	416: 100% 50/50 [00:20<00:00, 2.40it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.50it/s]
	all	91	536	0.916	0.933	0.948	0.544
Epoch 298/949	1.86G	0.02995	0.03287	0	0.06382	39	416: 100% 50/50 [00:20<00:00, 2.40it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.60it/s]
	all	91	536	0.911	0.94	0.953	0.554
Epoch 299/949	1.86G	0.03049	0.03423	0	0.06473	58	416: 100% 50/50 [00:20<00:00, 2.39it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.51it/s]
	all	91	536	0.916	0.937	0.959	0.553
Epoch 300/949	1.86G	0.02918	0.02895	0	0.05813	46	416: 32% 16/50 [00:06<00:14, 2.41it/s]

▼ Evaluate Custom YOLOv5 Detector Performance

Training losses and performance metrics are saved to Tensorboard and also to a logfile defined above with the `--name` flag when we train. In our case, we named this `yolov5s_results`. (If given no name, it defaults to `results.txt`.) The results file is plotted as a png after training completes.

Note from Glenn: Partially completed `results.txt` files can be plotted with `from utils.utils import plot_results; plot_results()`.

Disk 31.43 GB kullanılabilir

Yürütülüyor (1 sa. 53 dk. 45 sn.) Cell > run_cell_magic() > time() > <lambda> > time() > <module> > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()

Şekil 25. Eğitim Aşaması

Roboflow-Train-YOLOv5 adlı not defterinin kopyası

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Kaydediliyor...

Dosyalar

- sample_data
- test
- train
- valid
- yolov5
 - README.dataset.txt
 - README.roboflow.txt
 - data.yaml

Epoch	gpu_mem	box	obj	cls	total	targets	img_size
799/949	1.86G	0.02204	0.02537	0	0.04741	34	416: 100% 50/50 [00:21<00:00, 2.37it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.45it/s]
	all	91	536	0.932	0.944	0.958	0.554
Epoch 800/949	1.86G	0.02166	0.02574	0	0.0474	21	416: 100% 50/50 [00:21<00:00, 2.38it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.54it/s]
	all	91	536	0.932	0.944	0.958	0.555
Epoch 801/949	1.86G	0.02189	0.02369	0	0.04558	32	416: 100% 50/50 [00:21<00:00, 2.38it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.50it/s]
	all	91	536	0.932	0.942	0.958	0.555
Epoch 802/949	1.86G	0.0211	0.02461	0	0.04571	67	416: 100% 50/50 [00:20<00:00, 2.38it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.54it/s]
	all	91	536	0.935	0.94	0.958	0.555
Epoch 803/949	1.86G	0.02129	0.02531	0	0.0466	27	416: 100% 50/50 [00:20<00:00, 2.39it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:1.95: 100% 3/3 [00:01<00:00, 2.52it/s]
	all	91	536	0.937	0.938	0.958	0.554
Epoch 804/949	1.86G	0.02148	0.02495	0	0.04643	355	416: 80% 40/50 [00:16<00:04, 2.36it/s]

▼ Evaluate Custom YOLOv5 Detector Performance

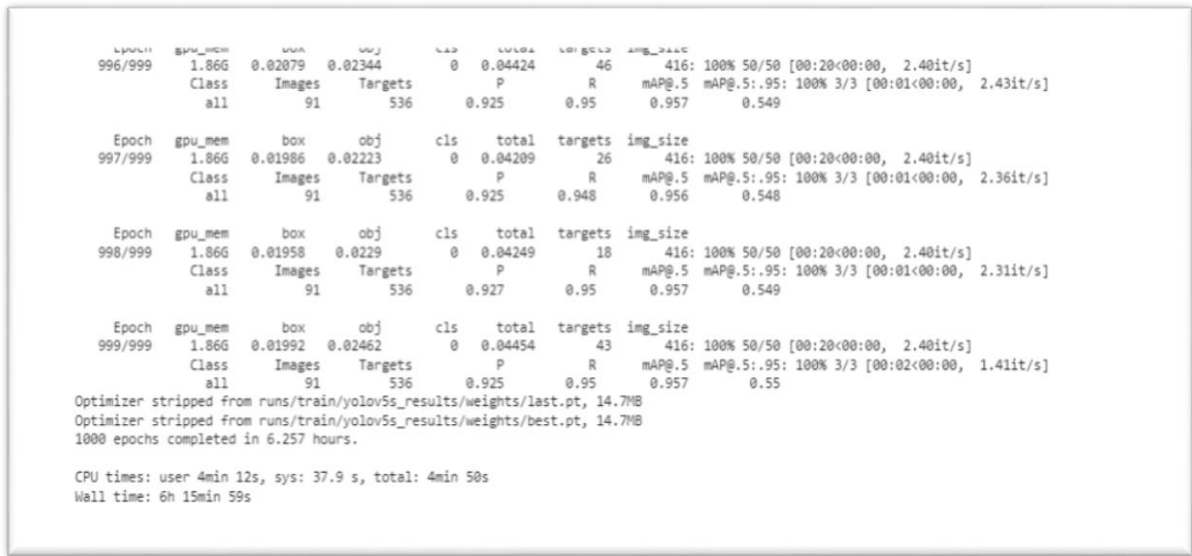
Training losses and performance metrics are saved to Tensorboard and also to a logfile defined above with the `--name` flag when we train. In

Şekil 26. Eğitim Aşaması 2

5. BULGULAR VE TARTIŞMA

Yapılan bütün deneyler, Intel(R) Core(TM) i7-6700HQ CPU @ 3.5 GHz, Intel(R) HD Graphics 530-NVIDIA GeForce GTX 950M (4GB), 12 RAM donanım gücüne sahip bilgisayarda, Python 3.8 Spyder Notebook geliştirme ortamı kullanılarak geliştirilmiştir.

Görüntüleri Roboflow ile etiketledikten sonra, DarkNet Mimarisi aracılığı ile Google Colab üzerinden 1000 iterasyon ile eğitime başlanmış ve 6 saat 15 dakikada kendi YOLO modelimiz eğitilmiştir. Eğitim sonucunda weight uzantılı ağırlık dosyalarımız oluşturulmuştur.



```
Epoch 996/999  gpu_mem 1.86G  box 0.02079  obj 0.02344  cls 0  total 0.04424  targets 46  img_size 416: 100% 50/50 [00:20<00:00, 2.40it/s]
Class Images Targets P R mAP@.5 mAP@.5:.95: 100% 3/3 [00:01<00:00, 2.43it/s]
all 91 536 0.925 0.95 0.957 0.549

Epoch 997/999  gpu_mem 1.86G  box 0.01986  obj 0.02223  cls 0  total 0.04209  targets 26  img_size 416: 100% 50/50 [00:20<00:00, 2.40it/s]
Class Images Targets P R mAP@.5 mAP@.5:.95: 100% 3/3 [00:01<00:00, 2.36it/s]
all 91 536 0.925 0.948 0.956 0.548

Epoch 998/999  gpu_mem 1.86G  box 0.01958  obj 0.0229  cls 0  total 0.04249  targets 18  img_size 416: 100% 50/50 [00:20<00:00, 2.40it/s]
Class Images Targets P R mAP@.5 mAP@.5:.95: 100% 3/3 [00:01<00:00, 2.31it/s]
all 91 536 0.927 0.95 0.957 0.549

Epoch 999/999  gpu_mem 1.86G  box 0.01992  obj 0.02462  cls 0  total 0.04454  targets 43  img_size 416: 100% 50/50 [00:20<00:00, 2.40it/s]
Class Images Targets P R mAP@.5 mAP@.5:.95: 100% 3/3 [00:02<00:00, 1.41it/s]
all 91 536 0.925 0.95 0.957 0.55

Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.7MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.7MB
1000 epochs completed in 6.257 hours.

CPU times: user 4min 12s, sys: 37.9 s, total: 4min 50s
Wall time: 6h 15min 59s
```

Şekil 27. Eğitim Sonucu

YOLO'nun çalıştırılabilmesi için Darknet Mimarisi bazı dosyalar beklemektedir. Bunlardan bir tanesi de .cfg dosyasıdır. Config dosyası içerisinde, YOLO içerisindeki Sinir ağının başarısını, hızını vs. etkileyecek özellikler yer almakta olup ve eğitim sırasında bu dosya eğitime göre düzenlenerek sisteme yüklenmiştir.

YOLO'nun çalışması için gerekli olan başka dosya ise. names dosyasıdır ve bu dosya içerisinde sınıfların isimleri tutulmaktadır. Güve tespiti uygulamamızda 1 sınıf olduğundan names dosyası bir sınıfa göre düzenlenmiştir.

Bir eğitimin başarısı, eğitim sonrası oluşan kayıp (loss) grafiğinden anlaşılabilmektedir. Eğer YOLO modelimiz için veri öğrenmesi ve yapay sinir ağımızın iterasyon sayısını çok fazla arttırsak sistem Overtraining (Aşırı Öğrenme) olabilmekte ve bu ise istemediğimiz bir durumdur.

Eğitimde 1000 iterasyon sonucunda;

Train/box_loss 0.02035

Train/obj_loss 0.02307

Map_0.5 (Ortalama Hassasiyet) 0.9579

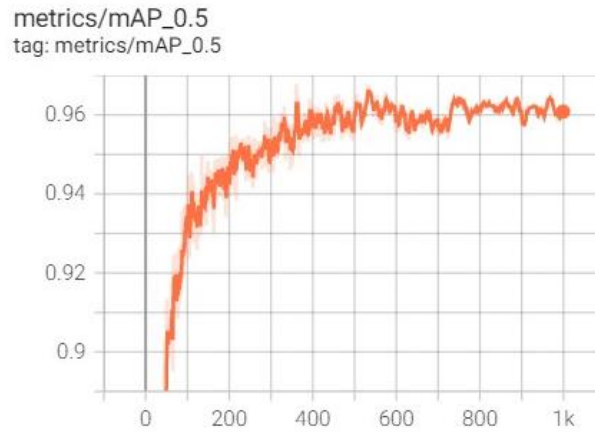
Map_0.5-0.95 (Lou) 0.5561

Precision 0.9256

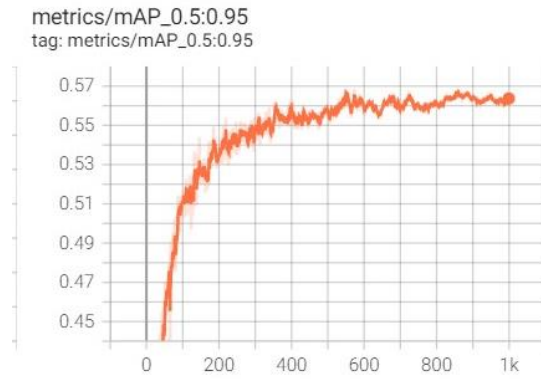
Recall 0.9496

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

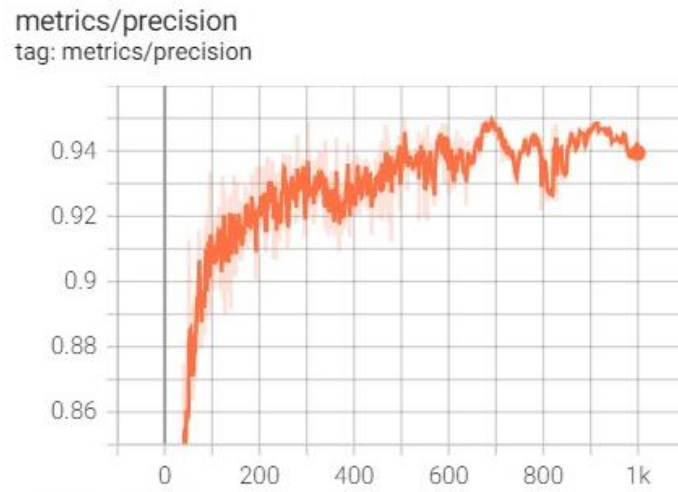
F1= 0.937



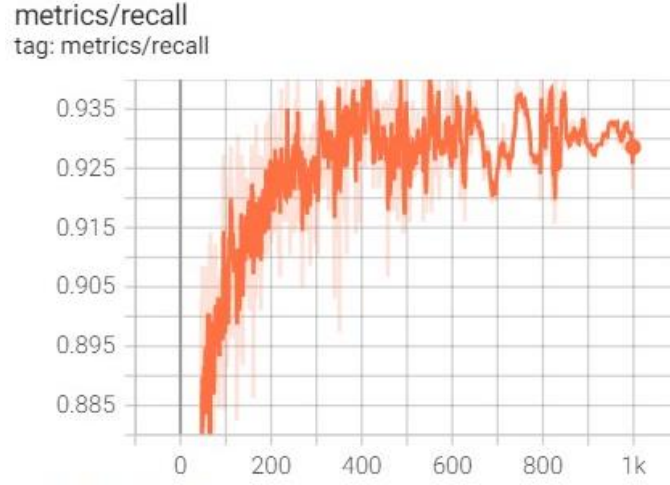
Şekil 28. Yolov5s tensorboard sonuçları



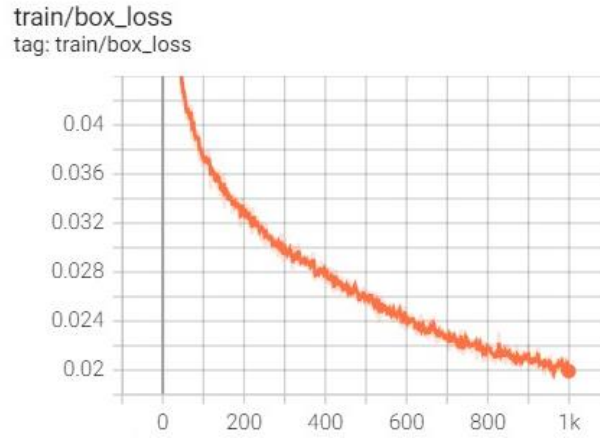
Şekil 29. Yolov5s tensorboard sonuçları 2



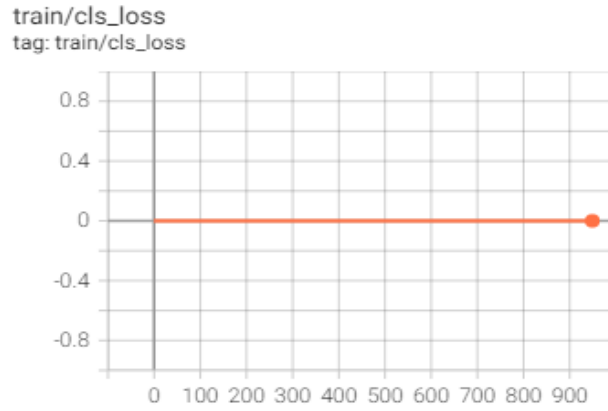
Şekil 30. Yolov5s tensorboard sonuçları 3



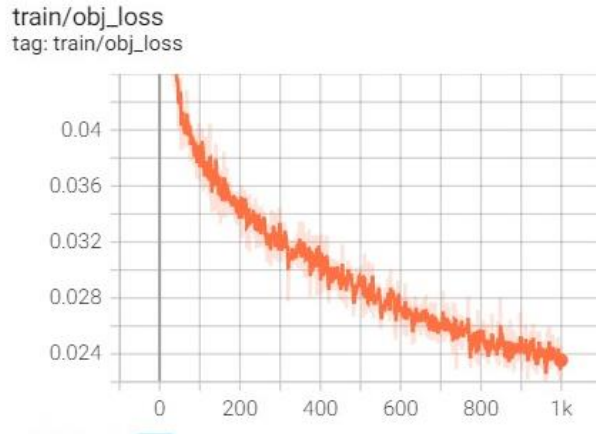
Şekil 31. Yolov5s tensorboard sonuçları 4



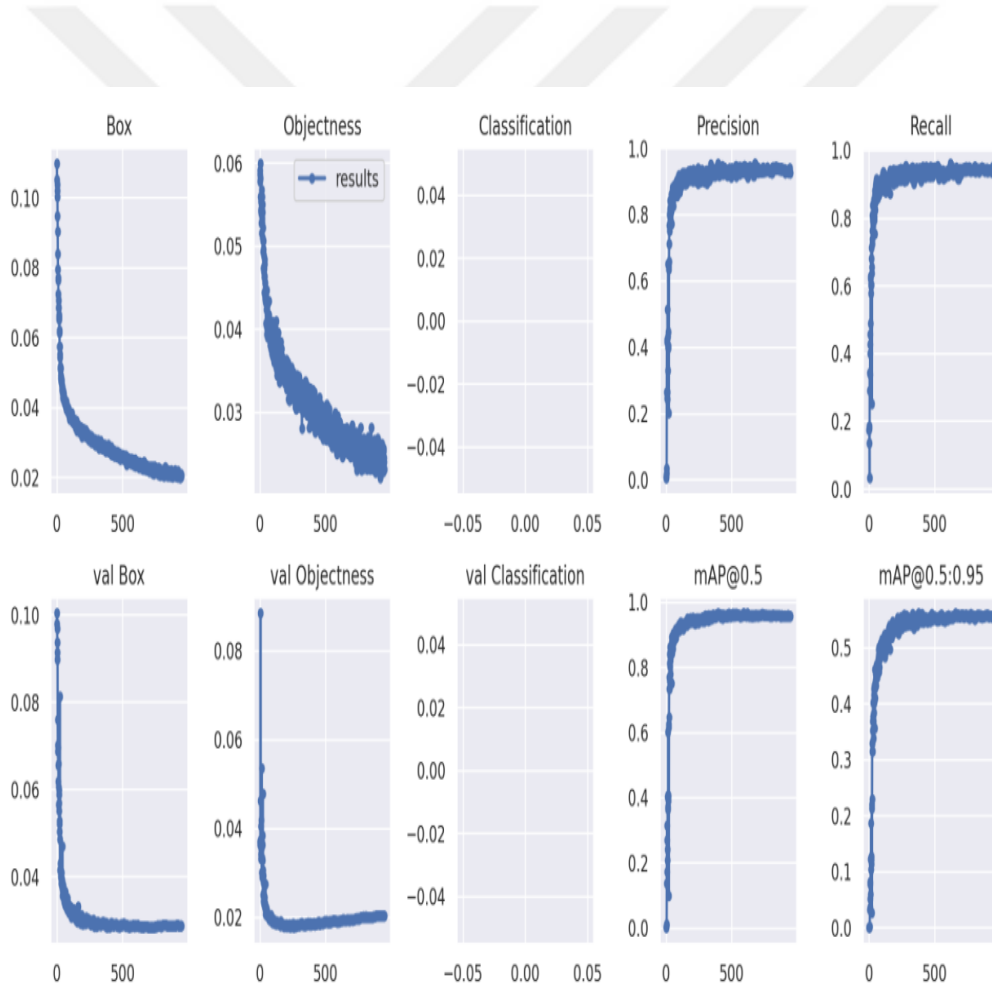
Şekil 32. Yolov5s tensorboard sonuçları 5



Şekil 33. Yolov5s tensorboard sonuçları 6



Şekil 34. Yolov5s tensorboard sonuçları 7



Şekil 35. Yolov5s tensorboard sonuçları 8

5.1. Kullanılan Yöntemler

5.1.1. Duyarlılık, Kesinlik, mAP ve F1 skor

Olasılık modeli kullanan sistemlerin genelinde, sisteme ait örnekler dört ayrı başlık altında toplanmıştır [20]. Bu başlıklar, yanlış pozitif (FP), doğru pozitif (TP), doğru negatif (TN) ve yanlış negatif (FN) değerlerinden oluşmaktadır. Bu değerler, kullanıcı tarafından etiketlenmiş verilerin etiket değerleri ve eğitilmiş sistem tarafından yapılan tahmin sonuçlarının uyuşup uyuşmaması durumuna göre oluşmaktadır. Karmaşıklık Matrisini Tablo.5’te gösterilmiştir.

Tablo 5. Karmaşıklık Matrisi

	Etelenmiş Durum		
	Toplam Durum	Pozitif Etiket	Negatif Etiket
Tahmin Edilen Durum	Pozitif Tahmin	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
	Negatif Tahmin	Yanlış Negatif (FN)	Doğru Negatif (TN)

Kesinlik (Precision): Kesinlik, yapılan doğru pozitif tahminlerin sayısını ölçen bir ölçümdür.

Doğru tahmin edilen pozitif örneklerin, tahmin edilen toplam pozitif örnek sayısına oranı olarak hesaplanır. Formülü ise Denklem 12’de görülmektedir.

$$P = \frac{TP}{TP+FP} \quad [12]$$

Duyarlılık (Recall): Tahmin edilen örneklerdeki doğru pozitif durumların, tüm pozitif sonuçlara oranıdır. Mümkün olduğunca yüksek olması gereklidir. Formülü ise Denklem 13’de görülmektedir.

$$R = \frac{TP}{TP+FN} \quad [13]$$

F1-skoru: Yukarıda yer alan Duyarlılık ve Kesinlik oranlarının harmonik ortalaması olarak hesaplanmaktadır. Formülü Denklem 14’te görülmektedir.

$$F_1 \text{ Skoru} = 2 \frac{P \times R}{P+R} \quad [14]$$

Ortalama Averaaj Kesinliği (mAP): Duyarlılık, F1-skoru ve kesinlik gibi değerlerin tek bir noktadan değerlendirilmesi için oluşturulmuş metrik olup, mAP değerinin formülü Denklem 15’te görülmektedir.

$$mAP = \int_0^1 P(R) dR \quad [15]$$

5.1.2. Kayıp Fonksiyonları

YOLO modelinde, hücrelerin her biri birden fazla sınırlayıcı kutu üretmektedir. Doğru pozitif olan durumlarda, kayıp değerini hesaplamak için üretilmiş sınırlayıcı kutular içerisinde en yüksek IoU değerine sahip olan kutu seçilmelidir. Bu sayede düşük IoU değerine sahip sınırlayıcı kutular elenmekte ve daha yüksek IoU değerine sahip olan sınırlayıcı kutular üretilmektedir [22].

YOLO'da hata fonksiyonları üç temel ana başlıkta incelenmektedir:

- 1-) Konum kaybı: Tahmin edilen kutudaki yanlışlık miktarını,
- 2-) Sınıflandırma kaybı: Tahmin edilen nesnedeki yanlışlık miktarını,
- 3-) Güven kaybı: Izgaranın içinde nesne olup olmama durumunun ne kadar yanlış olduğunu, ifade etmekte olup, bu kayıpların hepsi genel kaybı etkileyen faktörlerdir.

Konum Kaybı

Geçerli ızgarada tahmin edilen sınırlayıcı kutunun konumu ve boyut ile ilgili konum kaybı değerini hesaplamakta olup başka bir deyişle nesne olma durumunu ifade etmektedir. Bu hesaplama, yalnızca nesneyi tespit eden kutu yani ızgara için hesaplanmaktadır [22]. Konum kaybı Denklemi 16' da görülmektedir.

$$kk = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_j})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_j})^2] \quad [16]$$

1_{ij}^{obj} : i. hücredeki j. sınırlayıcı kutu nesne algılamaktan sorumlu ise 1 değilse 0.

λ_{coord} : Sınırlayıcı kutu koordinatlarındaki kayıp değerinin ağırlığını artırır.

kk : Konum kaybını ifade etmektedir.

Sınıflandırma Kaybı

Sınıflandırma kaybında nesne tespit edildiğinde her bir sınıf için koşullu sınıf olasılıklarının kare hatası hesaplanmaktadır. Sınıflandırma kaybı Denklem 17'de görülmektedir.

$$sk = \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in \text{sınıflar}} (p_i(C) - \hat{p}_j(C))^2 \quad [17]$$

1_{ij}^{obj} : Hücre de nesne varsa 1 yoksa 0 olma durumunu ifade etmektedir.

sk : Sınıflandırma kaybını ifade etmektedir.

$\hat{p}_j(C)$: i hücresindeki c sınıfı için koşullu sınıf olasılığı

Güven Kaybı

Resmin hücrelerinde gezerken çoğu ızgara hiçbir nesne içermeyecek, arka plan olarak tespit edilen sınıfların çok fazla olmasına sebep olacaktır. Bu durum sınıflar arasında bir dengesizliğe yol açmaktadır. Bu durumu ortadan kaldırmak veya azaltmak içinse eğer ızgarada nesne yoksa aşağıdaki formülde de görüldüğü gibi bu kayıp belirli bir sabitle (noobj ağırlığı) ile çarpılır. (Genellikle bu ağırlık 0,5 olarak tanımlanır.)

Nesne olması durumu:

$$gk = \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} [(c_i - \hat{c}_j)^2] \quad [18]$$

1_{ij}^{obj} : Hücre de nesne varsa 1 yoksa 0 olma durumunu ifade etmektedir.

$\hat{c}_j$: i hücresinde j. kutunun güven değeri.

gk : Nesne olması durumundaki Güven kaybını ifade etmektedir.

tk : Nesne olmaması durumundaki güven kaybını ifade etmektedir.

Nesne olmaması durumu:

$$n = \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} [(c_i - \hat{c}_j)^2] \quad [19]$$

1_{ij}^{noobj} : 1_{ij}^{obj} tümleyenidir.

$\hat{c}_j$: i hücresinde j. kutunun güven değeri.

λ_{coord} : Arka planın tespit edilmesi durumunda kaybı azaltmak için kullanılmaktadır.

nk : Nesne olmaması durumundaki güven kaybını ifade etmektedir.

Toplam kayıp ise; Güven kaybı, sınıflandırma kaybı ve konum kaybının toplamıdır.

$$\begin{aligned} tk = & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2] + \\ & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_j})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_j})^2] \\ & + \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} [(c_i - \hat{c}_j)^2] \\ & + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} [(c_i - \hat{c}_j)^2] \\ & + \sum_{i=0}^{s^2} 1_{ij}^{obj} \sum_{c \in sınıflar} (p_i(c) - \hat{p}_j(c))^2 \end{aligned} \quad [20]$$

5.2. Test Görselleri



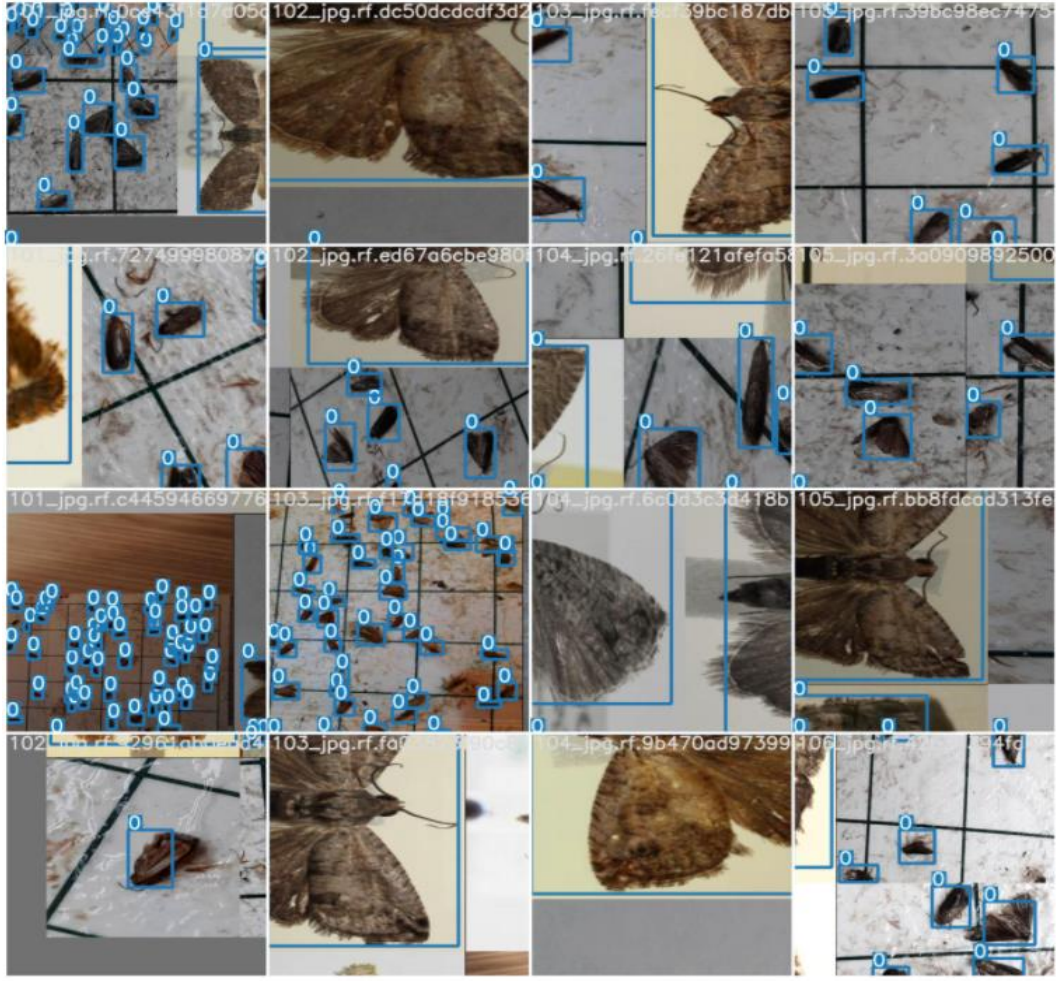
Şekil 36. Güve Nesnelerinin Tespiti



Şekil 37. Güve Nesnelerinin Tespiti 2.



Şekil 38. Tespit Edilen Nesnelerin Doğruluk Oranları



Şekil 39. Sınıfların Tespiti

6. SONUÇLAR

Nesne tespiti için tasarlanan yapay sinir ağı modelinin eğitim ve test süreçleri Google COLAB üzerinde gerçekleştirilmiştir. Eğitim sonrası elde edilen ağırlık dosyası Google Drive üzerinde yedeklenmiştir. Model yaklaşık olarak 6 saat eğitilmiştir. Eğitim işlemi 1000 iterasyon sürmüş ve ağırlıklar yedeklenmiştir. Yedeklenen bu ağırlıklar ile daha sonra test verileri üzerinde YOLO modeli çalıştırılmış ve ağırlığa ait ortalama hassasiyet (average precision) değerlerinin ortalamaları (Mean Average Precision) karşılaştırılmıştır. Eğitilen modelin performansını değerlendirmek için kullanılan metrik yöntemlerin (kesinlik (precision), yitim (loss), duyarlılık (Recall), ortalama hassasiyet (mAP), F1 skoru ve birleşim üzerinden kesişim (Intersection Over Union-IoU) sonuçlarına bakılmıştır.

Derin öğrenme ve yapay sinir ağları nesne tanıma algoritmaları geliştirilerek feromon tuzaklar içerisindeki nesnelerin tespiti, sayılması ve bahçenin ilaçlama zamanının tahmin edilmesi; görüntü, video ve gerçek zamanlı olarak kamera üzerinden tespitinin yapılması başarıyla sağlanmıştır. Bu çalışmada YOLO-V5 model ile güveler tespit edilebilmektedir. YOLO-V5 modelinde bulunan öznetelik katmanlarındaki düşük çözünürlüğe sahip görüntülerin iyileştirilmesi amaçlanmıştır. 5-katlamalı çapraz doğrulama uygulanarak elde edilen YOLOV5 modelimizin bahçe zararlısı doğu meyve güvesi algılama modelinin sonuç çıktıları;

IoU değeri %55,61, duyarlılık (recall) değeri %94.96, %93.7 F1-skor, %0.9579 ortalama hassasiyet değerleri elde edilmiş ve değerlere bakıldığında modelin çok iyi sonuç verdiği anlaşılmaktadır.

Train/box_loss 0.02035

Train/obj_loss 0.02307

Map_0.5 (Ortalama Hassasiyet) 0.9579

Map_0.5-0.95 (Lou) 0.5561

Precision 0.9256

Recall 0.9496

F1-skor 0.937

Gelecek çalışmalarda algoritmanın hata oranı düşürülebilir, daha büyük boyutlu girdilere daha doğru sonuç üretebilir hale getirilebilir ve tarım zararlısı çeşitliliğinin artırılarak algoritmanın daha fazla zararlıyı tanıyabilmesi sağlanabilir. Aynı zamanda hata oranını düşürmek amacıyla

farklı ağ yapılarının küçük objelerdeki performansının değerlendirilmesi, YOLO' ya bütünleştirilmesinin sağlanması ve öğrenme ve test aşamalarında ise, oluşturmuş olduğumuz veri seti genişletilerek, 5 fold cross validation ile de seçilimi daha da rastgele bir hale getirerek daha doğru sonuçlara ulaşılabilmesi sağlanabilir.



KAYNAKLAR

- [1] Ragıp SARI, İSPARTA ORMANLARINDA ZARARLI LEPIDOPTERA VE HYMENOPTERA TÜRLERİYLE MÜCADELEDE BİYOTEKNİK YÖNTEMLERİN KULLANIMI (İsparta, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2008), 55.
- [2] H. Yalçın, "Automatic inspection of pheromone traps," *2017 25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, 2017, pp. 1-4, doi: 10.1109/SIU.2017.7960429.
- [3] Mustafa Yılmaz, Görüntü İşleme Teknikleri İle Böcek Adedi Tespiti (İstanbul, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2016) 80.
- [4] Şenel, A., Tokat, S., ELECO , 29 Kasım – 1 Aralık 2012 , 2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyum, 2012, s613-617
- [5] Jin Kyo Jung, Bo Yoon Seo, Jum Rae Cho, Yong Kim, Monitoring of Mythimna separata Adults by Using a Remote-sensing Sex Pheromone Trap, Korean Journal of Applied Entomology Vol.52, pp.0-348, DOI : <https://doi.org/10.5656/KSAE.2013.10.0.058>
- [6] Karlos Espinoza, Diego L. Valera, José A. Torres, Alejandro López, Francisco D. Molina-Aiz, Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture, Computers and Electronics in Agriculture, Volume 127, 2016, Pages 495-505, ISSN 0168-1699, <https://doi.org/10.1016/j.compag.2016.07.008>
- [7] Lei, Yu, Yao, Zhifeng, He, Dongjian, (2018). Automatic detection and counting of urediniospores of Puccinia striiformis f. sp. tritici using spore traps and image processing. Scientific Reports. 8. 10.1038/s41598-018-31899-0.
- [8] Parmaksız, H., Karafil, A., & Özbay, H., Kesler, M., (2016). Farklı Eğim Açılarındaki Fotovoltaik Panellerin Elektriksel Ölçümlerinin Raspberry Pi ile İzlenmesi (The Monitoring of Photovoltaic Panels Electrical Measurements at Different Tilt Angles By Raspberry Pi)Düzce Üniversitesi Bilim ve Teknoloji Dergisi Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 4 (2016) 711-718.
- [9] Yüzgeç, U., & Aba, Ö., (2017). Raspberry Pi Kullanılarak bir Akıllı Ev Uygulaması Geliştirilmesi Development of a Smart Home Application using Raspberry Pi., Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, Cilt:4, Sayı:1, ISSN: 2458-7575 (<http://edergi.bilecik.edu.tr/index.php/fbd>)
- [10] Şahin, A , Çi l, S , İstanbullu, A . (2020). Raspberry Pi 4 ile Sürücü Yorgunluk Tespiti ve Uyarı Sistemi. Journal of Scientific, Technology and Engineering Research, 1 (1) , 13-18. DOI: 10.5281/zenodo.3902907
- [11] Özcan, Ömer & Yüce, Ali & Tan, Nusret. (2019). LabVIEW ve Raspberry Pi 3 Kullanarak Gerçek Zamanlı Sıcaklık Kontrol Uygulaması. 3rd International Mediterranean Science and Engineering Congress (IMSEC 2018) Çukurova University, Congress Center, October 24- 26, Adana / TURKEY Pages: 1-7, Paper ID:344
- [12] https://www.mathworks.com/solutions/deep-learning/convolutional-neural_network.html

Erişim Tarihi: 12.08.2021

[13] https://www.researchgate.net/figure/Software-with-corresponding-hardware-hierarchy-in-CUDA_fig2_323281068 Erişim Tarihi: 25.10.2021

[14] Wildml. "Understanding Convolutional Neural Networks ForNlp."(2016)

[15] <https://pjreddie.com/darknet/> Erişim Tarihi: 18.11.2021

[16] <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/> Erişim Tarihi: 08.09.2021

[17] https://tr.wikipedia.org/wiki/Raspberry_Pi Erişim Tarihi: 03.11.2021

[18] <https://www.technopat.net/> Erişim Tarihi: 22.07.2021

[19] <https://www.raspberrypi.com/> Erişim Tarihi: 25.03.2021

[20] <https://medium.com/deep-learning-turkiye> Erişim Tarihi: 03.08.2021

[21] Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. Paper presented at the Proceedings of the 27th European conference on Advances in Information Retrieval Research.

[22] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

ÖZGEÇMİŞ

İlk ve orta öğrenimini İstanbul’da tamamladı. 2006-2008 yılları arasında Kahramanmaraş Sütçü İmam Üniversitesi Bilgisayar Teknolojisi ve Programlama Bölümünden, 2009-2012 yılları arasında Anadolu Üniversitesi İşletme Fakültesi İşletme Bölümünden, 2014-2017 yılları arasında ise Yalova Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu. 2009-2012 yılları arasında özel sektörde, 2012 yılından beri Tarım ve Orman Bakanlığı’nda görev yapmakta olup, 2012-2014 yılları arasında İstanbul İl Tarım ve Orman Müdürlüğü’nde Veri Hazırlama ve Kontrol İşletmeni, 2014-2016 Yalova İl Tarım ve Orman Müdürlüğü’nde İdari ve Mali İşler Şube Müdürlüğü’nde Veri Hazırlama ve Kontrol İşletmeni, 2015-2019 yılları arasında Kırsal Kalkınma ve Örgütlenme Şube Müdürlüğü’nde Bilgisayar Teknikeri ve 2019 yılından beri aynı şubede Bilgisayar Mühendisi olarak çalışmaktadır.

TEZDEN TÜRETİLEN YAYIN VE ESERLER

[1] E. Tanrikulu, G. Harman. “Tarım Zararlılarının Derin Öğrenme ile Tespiti”, *7th INTERNATIONAL CONGRESS ON ENGINEERING AND TECHNOLOGY*, 2022, 16-18 Nisan, pp:677-686.