

Raspberry Pi ile Gerçek Zamanlı Bitki Algılama Uygulaması

Recai Özcan^{1*} ve Kemal Tütüncü²

¹Bozkır Meslek Yüksekokulu Elektrik ve Enerji Bölümü / Selçuk Üniversitesi, Türkiye

²Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü / Selçuk Üniversitesi, Türkiye

*(recaiozcan@selcuk.edu.tr)

Özet – Elektronik ve bilgisayar teknolojilerinin gelişimine paralel olarak yapay zekâ uygulamaları birçok alanda gelişme imkânı bulmuştur. Bu alanlardan birisi de hassas tarımda yapay zekanın kullanımınıdır. Yapay zekanın alt dalı olan derin öğrenme teknikleriyle güçlü donanıma sahip bilgisayarlar kullanılarak hassas tarım için birçok başarılı bilgisayarlı görü uygulamaları geliştirilmiştir. Ancak bu uygulamaların gerçek zamanlı çalışabilen bir robotik sisteme entegre edilmesi yüksek maliyet gerektirmektedir. Bu sebeple hassas tarıma yönelik gerçek zamanlı uygulamalar tasarlayabilmek ve robotik makinelerin alt sistemlerinde yapay zekayı kullanabilmek için düşük maliyetli çözümlere ihtiyaç duyulmaktadır. Bu çalışmada hassas tarımda robotik makinelerin bilgisayarlı görü sistemlerinde kullanmak için Raspberry Pi 4 ile gerçek zamanlı bir bitki algılama sistemi gerçekleştirilmiştir. Ayrıca sistemin algılama hızını artırmak için Coral USB hızlandırıcı kullanılarak sonuçlar algılama hızı bakımından değerlendirilmiştir. Coral USB hızlandırıcı ile Raspberry Pi 4'ün birlikte kullanımıyla 30 FPS'lere varan algılama hızı elde edilmiştir. Bu sonuçlar bir mikrobilgisayar üzerinde gerçek zamanlı bitki algılamanın yapılabileceğinin mümkün olduğunu göstermektedir.

Anahtar Kelimeler – Bitki Tespiti, Hassas Tarım, Raspberry Pi, Coral USB Hızlandırıcı, FPS

I. GİRİŞ

Bilgisayarlı görü, insan beyninin görmeye ilgili yapabileceği görevleri bilgisayar kullanarak yapmaya odaklı disiplinler arası bir alandır [1]. Başka bir deyişle bir kamera ve bilgisayar kullanılarak hedefleri tanımlamak için makinenin görmesini içeren bir teknolojidir ve temel olarak görüntü işleme, makine öğrenmesi ve derin öğrenmeyi içerir [2]. Özellikle, insan beyninden ilham alınarak bilgisayarlara verilerin nasıl işlendiğini öğretecek şekilde modellenen algoritmalar olan derin öğrenme [3] uygulamaları bilgisayarlı görüde yüksek verimlilikleri sebebiyle sıkça kullanılmaktadır.

Son yıllarda gerçekleştirilen bitki algılama uygulamalarının makine öğrenmesi yöntemlerinden uzaklaşarak, makine öğrenmesinin bir alt dalı olan derin öğrenme yöntemlerine doğru yöneldiği görülmektedir [4, 5]. Bunun nedeni makine öğrenimi yöntemlerinde verilerden özellik çıkarmak için uzman bilgisine bağlı olarak zaman alıcı ve

karmaşık bir süreç olan özellik mühendisliği bilgisi gerektirmesidir [4, 6]. Derin öğrenme yöntemleri ise veri özelliklerini otomatik olarak çıkararak temsili öğrenme yöntemini kullanmaktadır [3]. Yani derin öğrenme özellik çıkarma aşamasına gerek duymadan doğrudan verilerden öğrenebilir [7]. İnsan tahmin performansını aşan doğruluk seviyesine ulaşabilen [8] derin öğrenme bu özelliği sayesinde nesne algılamayı basitleştirmiştir [9].

Derin öğrenme alanında bir görüntüden bilgi çıkarmak için genel olarak evrimsel sinir ağları kullanılmaktadır [10]. Evrimsel sinir ağı tabanlı derin öğrenme algoritmaları kendine has ağ yapısı vasıtasıyla alt katmanlarda düşük seviyeli özellikleri, ilerleyen katmanlarda ise yüksek seviyeli özellikleri çıkarır [2] ve bu özelliklerden hangilerinin kullanılmasını gerektiğini kendisi belirleyerek tahminler üretir [11, 12].

Hassas tarım, tarımsal girdilerin ve çıktıların optimizasyonu [13] için bilişim çağının getirdiği ileri teknolojiyi tarımsal alanda kullanılmasına dayalı geliştirilen tekniklerdir [14]. Son yıllarda

gerçekleşen teknolojik gelişmelerle birlikte tüm dünyada uygulama alanı artmıştır [15]. Günümüzde bitki algılama alanında yenilikçi bir hassas tarım uygulaması geliştirebilmek ancak bilgisayarlı görü yaklaşımına dayalı yapay zeka çözümleriyle mümkün olmaktadır [16]. İlaveeten hassas tarımda bir bitki algılama tekniği geliştirebilmek ve gerçek zamanlı uygulamalarda kullanabilmek için düşük hesaplama gücüne sahip cihazlarda çalışabilen sistemler oluşturulması gerekmektedir [4].

Bu çalışmada, Raspberry Pi 4 ile gerçek zamanlı bir bitki algılama sistemi gerçekleştirilmiştir. Ayrıca sistemin algılama hızını artırmak için Coral USB hızlandırıcı kullanılarak sonuçlar değerlendirilmiştir.

II. MATERYAL VE YÖNTEM

A. Algılama Mimarisi

Nesne algılama, bilgisayarlı görü ile yakından ilgili bilgisayar teknikleri için kullanılan genel bir terimdir. Bu teknikte görüntüdeki bir nesnenin konumunu belirlenir ve belirli bir kategoride sınıflandırma yapılır. Burada amaç, belirli bir görüntüde var olan her nesneyi doğru bir şekilde bulmak ve mevcut sınıflardan birisine atamaktır [17]. Nesne algılama çerçevesinin bileşenleri genel olarak aşağıdaki gibi sıralanabilir.

1. Özellik Çıkarma: Bir görüntü, evrişimsel sinir ağları ile bir dizi üst düzey görsel özelliğe dönüştürülür.

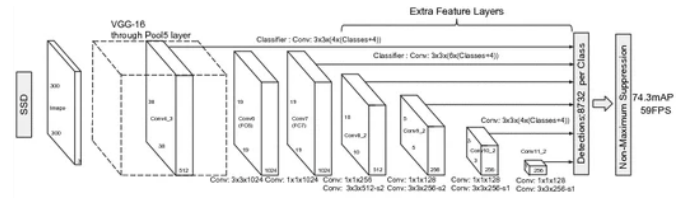
2. İlgi Bölgeleri: Model, nesne araması yapmak için görüntü alanlarını bulur. Yani algılamanın geçerli olup olmadığını değerlendirebileceği bir dizi sınırlayıcı kutu oluşturulur.

3. Yerelleştirme ve Sınıflandırma: Model, bu sınırlayıcı kutuların her biri için, içinde bir nesne olup olmadığını belirler ve mevcut sınıflardan birine güven puanı atayarak sınıflandırır. Ayrıca, bu adımda sınırlayıcı kutu koordinatlarının nesneyi mümkün olduğunca doğru bir şekilde çevrelemesi için ayarlama yapılır.

Nesne algılama alanında son teknoloji derin öğrenme algoritmaları genel olarak meta mimariler ve omurga mimarileri olmak üzere iki ana bölümden oluşmaktadır. Bir nesne tespit aşında, omurga mimarileri giriş görüntülerinden özellikler çıkarır. Meta mimariler ise çıkarılan özelliklere göre sınıflandırma yapar ve sınırlayıcı kutuların belirlenmesi işlemlerini gerçekleştirir.

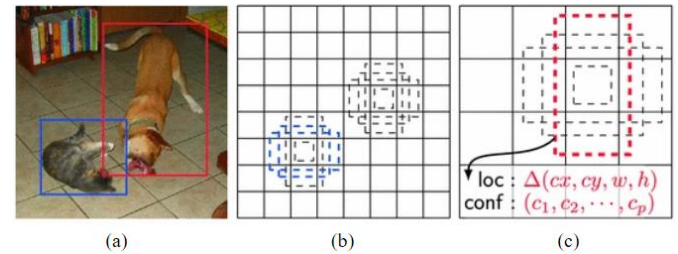
Bu çalışmada kullanılan SSD MobileNet v2, SSD (Single Shot MultiBox Detector) [18] meta

mimarisinin ve MobileNet v2 [19] omurga ağı ile oluşturulmuş bir nesne algılama ağıdır. Şekil 1'de SSD mimarisi görülmektedir.



Şekil 1. SSD mimarisi [18]

SSD mimarisinde girdi olarak bir görüntü alınır, bu görüntü evrişimsel sinir ağından geçirilir ve farklı boyutlarda özellik haritaları elde edilir. Tüm özellik haritalarında 3x3 evrişimsel filtre yardımıyla sınırlayıcı dikdörtgenler elde edilmektedir. Her dikdörtgen için aynı zamanda hem sınırlar hem de sınıflandırmalar belirlenmektedir. Bu dikdörtgenler her aktivasyon haritasında olduğu için hem küçük nesnelere hem de büyük nesnelere tanıyabilmektedir. Eğitim esnasında doğru olan sınırlar ile tahmin edilen sınırlar karşılaştırılmaktadır. En iyi tahmin yapan ve %50 oranının üzerindeki dikdörtgenler pozitif olarak etiketlenmektedir. Şekil 2'de SSD modelinin varsayılan sınırlayıcı kutu çerçevesi görülmektedir.



Şekil 2. SSD modelinin varsayılan sınırlayıcı kutu çerçevesi: (a) Gerçek kutular, (b) 8x8 özellik haritasında kutular, (c) 4x4 özellik haritasında kutular [18]

MobileNet v2, gömülü ve mobil görü uygulamaları için geliştirilen derinlemesine evrişimleri kullanan hafif bir ağ omurgası olan MobileNet v1'deki [20] evrişim katmanları arasına atlama bağlantılarını ekleyerek hesaplama maliyetini ve doğruluğunu artırmak için tasarlanmıştır.

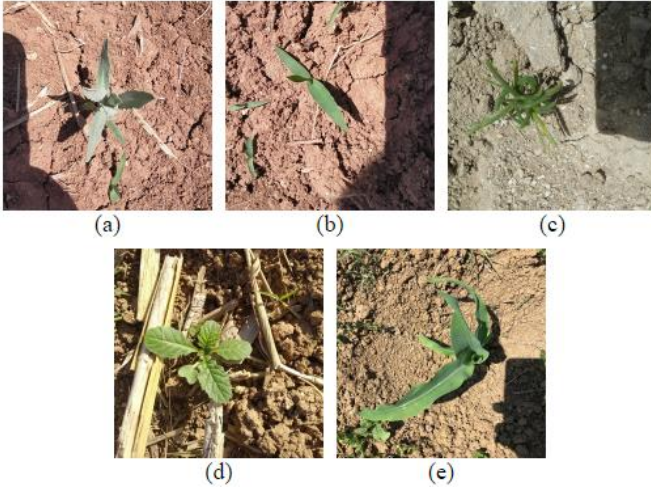
B. Veri Seti

Konya ve ilçelerinde cep telefonu ve fotoğraf makinesi kullanılarak, Tablo 1'de verilen ve Şekil 3'te örnek görüntüleri yer alan türlerden 500'er adet görüntü alınarak veri seti oluşturulmuştur. Bu görüntüler 640x640 piksel olacak şekilde yeniden

boyutlandırılmıştır. Her türden toplam 2500 adet görüntü model eğitim sürecinde %80'i eğitim için, %20 'si ise değerlendirme için kullanılmıştır.

Tablo 1. Veri setindeki türler ve sayılar

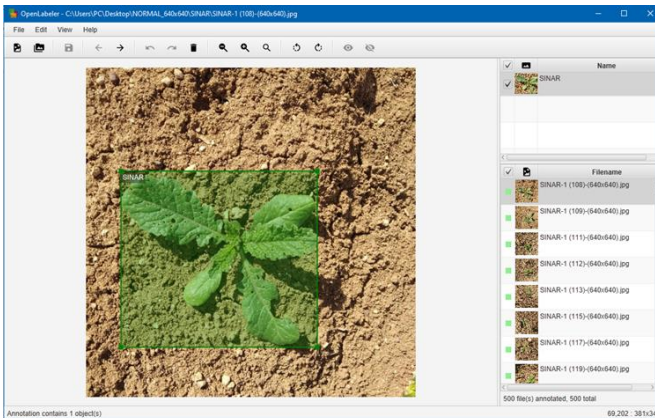
Bitki türleri	EPPO kodu	İmge sayısı	Eğitim	
			Eğitim	Değerlendirme
<i>Rhaponticum repens</i> (L.) Hidalgo	CENRE	500	400	100
<i>Panicum miliaceum</i> L.	PANMI	500	400	100
<i>Salsola kali</i> L.	SASKA	500	400	100
<i>Sinapis arvensis</i> L.	SINAR	500	400	100
<i>Zea mays</i>	ZEAMX	500	400	100
	Toplam	2500	2000	500



Şekil 3. Örnek veri seti görüntüleri (a) ve (f) *Rhaponticum repens* (L.) Hidalgo (CENRE); (b) ve (g) *Panicum miliaceum* L. (PANMI); (c) ve (h) *Salsola kali* L. (SASKA); (d) ve (i) *Sinapis arvensis* L. (SINAR); (e) ve (j) *Zea mays* (ZEAMX)

C. Veri Etiketleme

Etiketleme işlemleri için Şekil 4'te örnek görüntüsü verilen açık kaynak kodlu OpenLabeler [21] aracı kullanılarak tüm resimler etiketlenerek xml dosyaları oluşturulmuştur.



Şekil 4. OpenLabeler uygulamasının örnek görüntüsü

D. Model Eğitimi

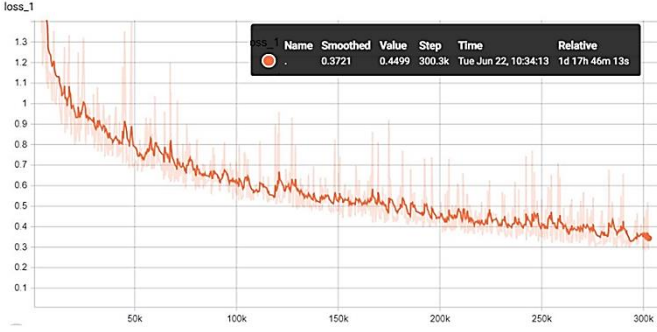
Sıfırdan model oluşturmak ve modeli eğitmek için çok fazla zaman ve güçlü bilgisayarlara gereksinim duyulacağından, imkanlar dahilinde bir görev için önceden eğitilmiş bir modelin ikinci bir görevde yeniden tasarlandığı yöntem olan transfer öğrenimi yöntemiyle model oluşturulmuştur. Modelde Google tarafından geliştirilen bir derin öğrenme kütüphanesi olan ve açık kaynak kodlu olarak sunulan Tensorflow [22] kullanılmıştır.

Tercih edilen model önceden Tensorflow Object Detection API ile 91 nesne türü ve 2,5 milyon etiketli veriden oluşan Common Objects in Context (COCO) [23] veri seti kullanılarak eğitilmiştir. Modeli kendi oluşturduğumuz veri setini kullanarak transfer öğrenimi yöntemiyle tekrar eğitip, nesne tespiti yapılmıştır. Böylece az bir veriyle düşük donanımlı bir bilgisayar üzerinde eğitim gerçekleştirilerek başarı sonuçları gözlemlenebilmiştir.

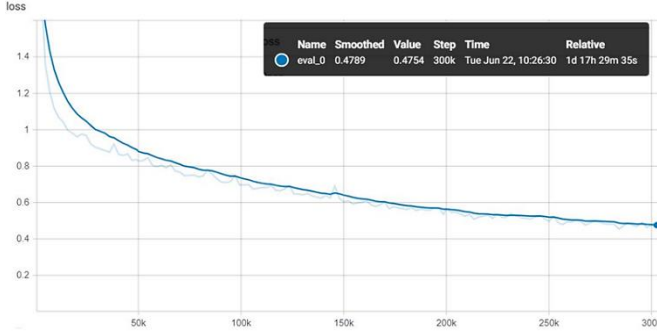
Önceden eğitilmiş modeller kullanılarak transfer öğreniminde yapılanlar:

- Konvolüsyon katmanlarının katsayıları sabitlendi,
- Çıkış katmanındaki aktivasyon fonksiyonu ile aktive olan çıkış nöronlarının sayısı istenilen sınıf sayısı ile aynı olacak şekilde değiştirildi,
- Çıkış katmanı ile konvolüsyon katmanları arasındaki Fully Connected Layer'a ait ağ katsayıları başlangıç değerleri rastgele olacak şekilde yeni verilerle tekrar eğitildi.

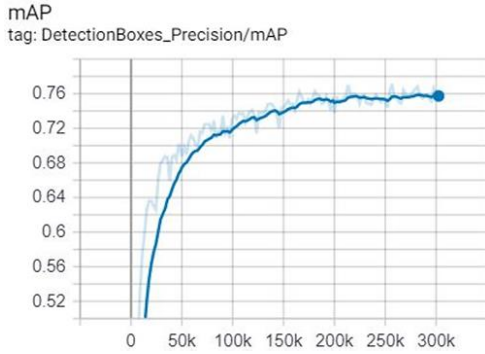
Oluşturulan veri seti ile eğitim işlemi Intel Core i7-10750H CPU ve NVIDIA GeForce RTX 2060 GPU üzerinde gerçekleştirilmiştir. Eğitimde ADAM optimizör kullanılmıştır. Konfigürasyon dosyasında başlangıç öğrenme oranı 0-4500 adım 0.0002, 4500-7000 adım 0.0001, 7000-10000 adım 0.00008 ve 10000. adımdan sonra 0.00004 olarak belirlenerek 30000 adımda model eğitimi gerçekleştirilmiştir. Eğitim işlemlerine ait parametreler ve grafikleri Şekil 5, Şekil 6 ve Şekil 7'de görülmektedir.



Şekil 5. Eğitim grafiği



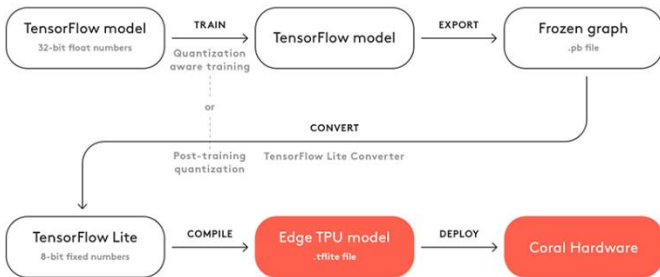
Şekil 6. Değerlendirme grafiği



Şekil 7. Eğitimin nesne tespit kutuları için ortalama duyarlılık (mean average precision (mAP)) grafiği

E. Model Dönüşüm İşlemleri

Eğitim işlemlerinden sonra Şekil 8’de görüldüğü gibi modelin Raspberry Pi ve Coral USB hızlandırıcı üzerinde çalışabilmesi için model dönüşüm işlemleri gerçekleştirilmiştir.



Şekil 8. Model dönüşümü için temel iş akışı [24]

Ağ içerisinde değerler 32 bit float biçimindedir. Bu değerler düşük donanımlı cihazların ve Coral

USB hızlandırıcının çalıştırabileceği 8-bit integer değerlere dönüştürülmüştür. Modelin Raspberry Pi ve Coral üzerinde çalışabilmesi için model doğrudan Tensorflow Lite’a dönüştürülebilir veya istenirse Frozen graph (model bu şekilde bilgisayar üzerinde çalışabilir) şeklinde dışarıya aktarılabilir ve bu dosya Tensorflow Lite’a dönüştürülerek Raspberry Pi de kullanılabilir. Tensorflow Lite’a dönüştürülen dosya Edge TPU model’a derlenerek Coral USB hızlandırıcı üzerinde çalışabilmesi sağlanabilir.

F. Deneysel Kurulum

Eğitilen modeli Raspberry Pi üzerinde yüksek FPS’te (saniyede alınan görüntü sayısı) çalıştırmak ve çıkarım hızını artırmak için Coral USB Hızlandırıcı kullanılmıştır. Coral USB Hızlandırıcı, özel olarak tasarlanmış bir uygulamaya özgü entegre devre (ASIC) çözümüdür. Raspberry Pi’nin USB bağlantı noktasına bağlanarak sisteme bir Edge TPU yardımcı işlemcisi eklemek suretiyle yüksek hızlı makine öğrenimi çıkarımları sağlar. Saniyede 4 trilyon aritmetik işlem yapabilme kabiliyetine sahiptir. Oluşturulan sistemde kullanılan Raspberry Pi HQ Kamera ve Raspberry Pi 6mm Geniş Açık Lens Raspberry Pi ile kullanılabilen diğer kamera modüllerine kıyasla çok daha net ve yüksek kalitede çekim yapabilir. 12 MP ve HD 30 FPS çekim yapabilme yeteneğine sahiptir. Raspberry Pi 6mm Geniş Açık Lens ile manuel odaklama yapılarak belirli mesafedeki nesnelerin net görüntüsü alınabilir. Şekil 9’da sistemin donanım elemanları görülmektedir.



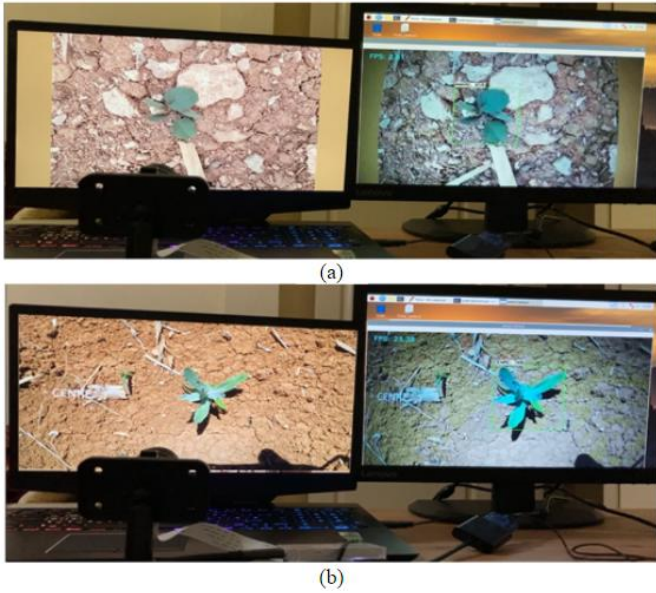
Şekil 9. Sistemin donanım elemanları

Oluşturulan sistemde yazılım ve donanımın birbirleriyle uyum içerisinde çalışabilmesi için birçok deneme yapılarak uygun kütüphane, sürücü ve programlar belirlenerek kurulumları

gerçekleştirilmiştir. Oluşturulan sistemin yazılım kısmında Python 3.7.9 programlama dili, Tensorflow 1.14 ve OpenCV-Python kütüphaneleri kullanılmıştır.

III. BULGULAR

Gerçek zamanlı test işlemleri için modelin daha önce hiç görmediği görüntülerden bir video görüntü hazırlanmıştır. Bu işlem için iki monitor kullanılarak gerçekleştirilen uygulamada sadece Raspberry Pi kullanılarak sistemin çalışması Şekil 10 (a)'da ve Raspberry Pi + Coral USB Hızlandırıcı çalışması Şekil 10 (b)'de görülmektedir. Solda video görüntü üzerinde algılacak olan nesne görüntüsü, sağda Raspberry Pi'ye bağlı olan monitor üzerinde modelin algılaması görülmektedir.



Şekil 10. Oluşturulan sistemin testleri (a) Sadece Raspberry Pi kullanılarak sistemin çalışması, (b) Raspberry Pi + Coral USB Hızlandırıcı çalışması

Hazırlanan video görüntü üzerinde Raspberry Pi 4 kullanılarak veya Raspberry Pi 4 + Coral USB hızlandırıcı kullanılarak testler gerçekleştirilmiştir. Sadece Raspberry Pi 4 kullanıldığında 2-3 FPS (saniye başına görüntü sayısı), Raspberry Pi ve Coral USB hızlandırıcı birlikte kullanıldığında ise 20-30 FPS arasında algılama hızları elde edilmiştir.

IV. TARTIŞMA

Bu sonuçlardan Raspberry Pi 4'ün tek başına kullanımının hassas tarımda gerçek zamanlı uygulamalar için yeterli olmadığı, ancak Coral USB hızlandırıcı ile Raspberry Pi 4'ün birlikte kullanımının umut vadeci olduğu görülmektedir. Nitekim algılama hızı Raspberry Pi 4'ün tek başına

kullanımına nazaran en az 10 kat artırdığı açıkça görülmüştür.

V. SONUÇLAR

Bu çalışmada, yapay zekanın hassas tarım uygulamalarında kullanılabilmesi için düşük maliyetli ve düşük güçlü bilgisayarlarda kullanımının mümkün olup olmadığının araştırması yapılmıştır. Bunun için son teknoloji bir derin öğrenme modeli transfer öğrenimi yöntemiyle eğitilmiş, gerekli donanım ve yazılımların bir araya getirilmesiyle bu modelin Raspberry Pi 4 ve Coral USB hızlandırıcı üzerinde çalışabilmesi sağlanmıştır. Elde edilen sonuçlara göre bir mikrobilgisayar üzerinde yapay zekanın hassas tarım uygulamaları için kullanılan robotik makinelerin alt sistemlerinde kullanılabileceği kanıtlanmıştır.

Bu çalışmada oluşturulan sistemin FPS bakımından değerlendirmesi yapılmıştır. Çalışmanın yazarları Raspberry Pi ile çeşitli derin öğrenme modelleri eğiterek modellerin başarımlarını yönünden de değerlendirmek için ek çalışmalar gerçekleştireceklerdir.

KAYNAKLAR

- [1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," (in English), *Comput Intel Neurosc*, vol. 2018, 2018, doi: 10.1155/2018/7068349.
- [2] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation —A review," (in en), *Information Processing in Agriculture*, vol. 7, no. 1, pp. 1-19, March 1, 2020 2020, doi: 10.1016/j.inpa.2019.09.006.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," (in en), *Nature*, vol. 521, no. 7553, pp. 436-44, May 28 2015, doi: 10.1038/nature14539.
- [4] A. S. M. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, and M. G. K. Jones, "A survey of deep learning techniques for weed detection from images," (in English), *Computers and Electronics in Agriculture*, vol. 184, p. 106067, May 2021, doi: 10.1016/j.compag.2021.106067.
- [5] B. Liu and R. Bruch, "Weed Detection for Selective Spraying: a Review," (in en), *Current Robotics Reports*, vol. 1, no. 1, pp. 19-26, 2020-03-01 2020, doi: 10.1007/s43154-020-00001-w.
- [6] J. Amara, B. Bouaziz, and A. Algergawy, *A Deep Learning-based Approach for Banana Leaf Diseases Classification*. BTW workshop, Stuttgart, pp.79-88 (in en), 2017.
- [7] P. Wspanialy, J. Brooks, and M. Moussa, "An Image Labeling Tool and Agricultural Dataset for Deep Learning.," *ArXiv Preprint ArXiv:2004.03351*, 2020.

- [8] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, no. 9, pp. 2352-2449, Sep 2017, doi: 10.1162/NECO_a_00990.
- [9] W. Z. Zhao and S. H. Du, "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach," (in English), *Ieee Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544-4554, Aug 2016, doi: 10.1109/Tgrs.2016.2543748.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012 2012, vol. 25, pp. 1097-1105.
- [11] A. Milioto, P. Lottes, and C. Stachniss, "Real-Time Blob-Wise Sugar Beets Vs Weeds Classification for Monitoring Fields Using Convolutional Neural Networks," (in English), *Isprs Ann Photo Rem.*, vol. 4-2, no. W3, pp. 41-48, 2017, doi: 10.5194/isprs-annals-IV-2-W3-41-2017.
- [12] J. Fuentes-Pacheco *et al.*, "Fig Plant Segmentation from Aerial Images Using a Deep Convolutional Encoder-Decoder Network," (in English), *Remote Sensing*, vol. 11, no. 10, p. 1157, May 2 2019, doi: 10.3390/rs11101157.
- [13] D. J. Mulla, "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps," (in English), *Biosystems Engineering*, vol. 114, no. 4, pp. 358-371, Apr 2013, doi: 10.1016/j.biosystemseng.2012.08.009.
- [14] U. Türker *et al.*, "Hassas Tarım Teknolojilerindeki Gelişmeler," *Türkiye Ziraat Mühendisliği VIII. Teknik Kongresi Bildiriler Kitabı-1*, p. 295, 2015.
- [15] M. Akkamış and S. Çalışkan, "İnsansız Hava Araçları ve Tarımsal Uygulamalarda Kullanımı," *Türkiye İnsansız Hava Araçları Dergisi*, vol. 2, no. 1, pp. 8-16, 2020.
- [16] K. Sudars, J. Jasko, I. Namatevs, L. Ozola, and N. Badaukis, "Dataset of annotated food crops and weed images for robotic computer vision control," (in en), *Data Brief*, vol. 31, p. 105833, Aug 2020, doi: 10.1016/j.dib.2020.105833.
- [17] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," (in en), *Comput Biol Med*, vol. 128, p. 104129, Jan 2021, doi: 10.1016/j.compbiomed.2020.104129.
- [18] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," (in English), *Lect Notes Comput Sc.*, vol. 9905, pp. 21-37, 2016, doi: 10.1007/978-3-319-46448-0_2.
- [19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018 2018, pp. 4510-4520.
- [20] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.," *arXiv preprint arXiv:1704.04861*, 2017-04-16 2017.
- [21] K. H. Wong. "OpenLabeler." <https://github.com/kinhong/OpenLabeler> (accessed 5 December, 2022).
- [22] Tensorflow. "TensorFlow 1 Detection Model Zoo." https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md (accessed 12 November, 2022).
- [23] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014 2014, Cham: Springer International Publishing, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.
- [24] Google Coral. "Tensorflow model overview." <https://coral.ai/docs/edgetpu/models-intro/#compatibility-overview> (accessed 16 December, 2022).