

**DERİN ÖĞRENME TEMELLİ MASKELİ YÜZ TANIMA
DEEP LEARNING BASED MASKED FACE DETECTION**

Ramazan Rahmetullah SAHİN

*Kutahya Dumlupınar University, Engineering Faculty, Electrical-Electronics Engineering
Department, Kutahya, Türkiye.*

ORCID ID: 0000-0002-6769-4858

Mustafa NAMDAR

*Kutahya Dumlupınar University, Engineering Faculty, Electrical-Electronics Engineering
Department, Kutahya, Türkiye.*

ORCID ID: 0000-0002-3522-4608

Arif BASGUMUS

*Bursa Uludag University, Engineering Faculty, Electrical-Electronics Engineering Department,
Bursa, Türkiye.*

ORCID ID: 0000-0002-0611-3220

Özet

Yüz tanıma teknolojisi, bir dinamik veya statik görüntüdeki insan yüzlerini tanımlamak için çalışan bir dizi algoritmadır. Son dönemde yazılım alanındaki gelişmeler ile yüz tanıma alanındaki çalışmalar hız kazanmış ve kullanımı artmıştır. Yüz tanıma teknolojisi, bir kişinin yüzünün belirli kısımlarını tarayarak ve ölçerek çıkardığı biyometrik verilerle çalışır ve fiziksel temas gerektirmediği için daha güvenlidir. Koronavirüs fiziksel temas yoluyla yayıldığından, en popüler kimlik tanıma sistemleri (parmak izi, parola) dahi güvensiz hale gelmiştir. Bu çalışmada, COVID-19 virüsünün bulaşmasından korunmak amacıyla takılan yüz maskelerinin, maske kullanırken derin öğrenme temelli yüz tanıma ve tespit yöntemleri sunulmaktadır. Görüntü analizi ile nesneleri gruplandırıp birbirlerinden ayıran Konvolüsyonel Sinir Ağları'ndan (CNN) yararlanmak için Tensorflow kütüphanesi kullanılmıştır. İçerisinde birçok derin öğrenme mimarisi ve yapay zeka modelleri bulunduran TensorFlow'un en önemli avantajlarından biri geliştirilen projelere uygun mimariyi sağlayabilmesidir.

Anahtar Kelimeler: Derin öğrenme, yüz tanıma, maske, Tensorflow

Abstract

Face recognition technology is an array of algorithms that work to recognize dynamic or static face images of people. Recently, with the developments in the field of software, studies in the field of face recognition have accelerated and its use has increased. Even most popular identification systems became insecure because the spread of coronavirus occurs through physical contact. In this study, deep learning based face recognition and detection methods are presented based on face masks, which be worn for being protected from COVID-19 epidemic. Tensorflow library is being used to benefit from CNN which are used to separate objects by analyzing images and grouping the common features. One of the most important advantages of Tensorflow is to provide a convenient library to developed projects.

Key words: Deep Learning, Face Recognition, Mask, Tensorflow

Giriş

Yüz tanıma, ekonomi, güvenlik, savunma sanayi, sağlık, vb. gibi birçok alanda, otonom hayata geçişte teknolojinin kullanıldığı en popüler biyometrik tanımlama yöntemlerinden biridir. Bu teknolojiyi hayata

geçirmekte makine öğrenmesi ve özellikle derin öğrenme algoritmaları yaygın olarak kullanılmaktadır. Derin öğrenme ile elde edilen başarılar, özellikle yüz tanıma alanında gelişimi, ilerlemeyi ve yaygın kullanımı beraberinde getirmiştir. 2019 yılı sonunda başlayıp günümüzde de farklı varyantları ile devam etmekte olan COVID-19 virüsünün, fiziksel temas yolu ile yayılmakta olduğu yapılan araştırmalar ile ortaya konulmuştur. Bu süreçte, maske kullanımının insan sağlığı açısından çok önemli olduğu gerçeği de vurgulanmıştır. Bu korunma yöntemi, doğru bir şekilde uygulanmadığında COVID-19 vakalarının artışına ve karantina sürecinden dolayı da birçok sektörün hizmetlerinin aksamasına yol açmaktadır. Bu durumda, maske kullanımının denetlenmesi, özellikle direkt olarak virüs tehdidi ile karşı karşıya olan sağlık çalışanları için zorunlu hale gelmiştir. Son dönemde maskeli yüz tanıma ve algılama ile ilgili çalışmalarda artış görülmektedir. [1]'de, imgeler üzerinden farklı modelleme yöntemleri kullanılarak yapılan çalışmada, yüz algılamada maske takıp takılmadığına dair %99,65, yüz tanımda maske takan kişilerde %99,52 ve takmayan kişilerde de %99,96 doğruluğa ulaşılmıştır. Çalışmada 13359 görsel kullanılmıştır. [2]'de Faster R-CNN kullanarak maskeli yüz algılamaya yönelik gerçekleştirilen çalışmada, yeterli veri seti sunulmadığı için %86 ile %57 arasında değişen doğruluk oranına ulaşılabilmektedir. [3]'teki çalışmada ise gerçek zamanlı maskeli yüz algılama sisteminin, Kütahya Dumlupınar Üniversitesi öğretim üyeleri ve öğrencilerinin gerçekleştirdiği [4]'teki çalışmada da kullanılan YOLO mimarisi ile gerçekleştirildiği görülmektedir. Burada, gerekli veri seti sağlanmış ve %99,5 oranında başarıya ulaşılmıştır. Bu çalışma, Faster R-CNN Inception V2 kullanarak maskeli yüzleri algılayıp maskesiz yüzlerin de tanınmasını gerçek zamanlı olarak gerçekleştirmektedir.

Çalışmamızda maske kullanımının algılanması, yanlış kullanım yapan ve maske takmayan kişilerin tespit edilmesi gerçek zamanlı olarak sağlanmıştır. CNN (Konvolüsyonel Sinir Ağı), mimarisinin avantajlarından yararlanılmıştır. Kütüphane ve modelleme yöntemlerinden, yüz tanıma için uygun olarak daha önceki çalışmalarda yeterli sonuçları vermiş yöntemler kullanılmıştır. Kamera üzerinden elde edilen veriler gerçek zamanlı olarak değerlendirildiğinden modellemelerin performansı da değişiklik göstermektedir. Bu modellemelerden biri Faster R-CNN Inception V2'dir. Faster R-CNN, birçok nesne tanımlama projelerinde kullanılmış başarılı bir modelledir. Faster R-CNN kullanılan projelerden biri de görme engelli bireylerin dış ortamda karşılarına çıkabilecek nesneleri ayırt etmeyi amaçlayan [5]'teki çalışmadır. Bu çalışma Kütahya Dumlupınar Üniversitesi öğretim üyeleri ve lisans öğrencileri tarafından geliştirilmiştir. Faster R-CNN metodunun Inception V2 mimarisi sayesinde yüksek doğruluk oranına sahip olması sağlanmıştır. Bu modelin içerisinde maskeli ve maskesiz yüz görsellerinin bulunduğu veri kümesi ile eğitilmesi sağlanmış ve veriler üç sete ayrılmıştır: maskeli yüz, maskesiz yüz ve maskesiz tanınan yüz.

Faster R-CNN modeli konvolüsyonel katmanlardan özellik haritalarını paylaşır. Pooling (havuzlama) katmanı yerine RoI (ilgi bölgesi) katmanını kullanır. Böylece paylaşım şeması hesaplama işleminden tasarruf sağlar. Aday bölgeleri bulma süresini azaltmak için RPN de (Bölge Öneri Ağı) kullanan Faster R-CNN modelinin, maskeli yüzleri tanıma ve tespit yöntemlerinde kısa zamanda en doğru ve kararlı sonucu verdiği bilinmektedir.

Inception V2'de V1'den farklı olarak kullanılan 5x5 matrisler, 3x3 boyutunda iki adet matrisle değiştirilmekte ve bu yolla hesaplama süresi azaltılarak 2,78 kat daha hızlı sonuç elde edilmektedir [6].

Çalışmanın ikinci bölümünde CNN mimarisi kullanılan dokümanların özellikleri, derin öğrenme algoritmasının aşamalara bölünmüş işleyişi ile ele alınmaktadır. Üçüncü bölümde tasarlanan çalışma için elde edilen performans analizi sonuçları ve hata senaryoları aktarılmaktadır. Çalışmada elde edilen önemli sonuçlar, muhtemel araştırma konuları ile birlikte sonuçlar bölümünde verilmektedir.

Yöntem

Çalışma için programlanan yapay zekâ algoritmasının, kamerada tespit edilen kişinin daha önce derin öğrenme yöntemi ile kendisine öğretilen yüzünü tanıyabilecek kabiliyette olması amaçlanmaktadır. Kameradan elde edilen görüntü matrisler halinde işlenerek sayısal veri haline getirilmekte ve bu verilerin programın belleğinde bulunan veri seti içerisinde eşleştirme yapılması sağlanmaktadır. Eşleştirme sonucunda, kişiler maskeli ve maskesiz olarak tespit edilmektedir.

CNN ve Derin Öğrenme

CNN (Konvolüsyonel Sinir Ağı), girdileri analiz ederek bunların ortak özelliklerini gruplandırıp, nesneleri birbirinden ayırmaya yarayan evrimsel bir ağıdır. CNN, bu ortak özellikleri gruplandırma işlemini katmanlarla sağlar. Bu katmanlardan birkaçı; konvolüsyon katmanı, ReLU katmanı ve RoI katmanı olarak sayılabilir. Derin öğrenme, yapay sinir ağlarını ve dolayısıyla onun katmanlarını kullanarak, giriş verisini belirli parametrelere göre işleyip analiz ederek öğrenip, otonom bir şekilde karar alan makine öğrenmesi türüdür. Makine öğrenmesinden farklı olarak insan beyninin çalışma mekanizmasına benzer stratejide çalışır. Makine öğrenmesi, kullanıcı girdisi sayesinde veriler arasındaki farka odaklanırken; derin öğrenme, bu algılamayı yapay sinir ağları ile gerçekleştirir. Giriş verisi nöronlar ve kanallar aracılığı ile gizli katmanlardan geçer. Gizli katmanlarda nöronların aktif olup olmadığı bilgisi aktivasyon fonksiyonları denen bazı matematiksel fonksiyonlar sayesinde edinilir. Algoritma, giriş verisinin 3x3 pikseller halinde denetlenmesi ile elde ettiği sayısal verileri öğrenir ve sonuca ulaşır [7].

Konvolüsyon Katmanı (Convolution Layer)

Bu katman, CNN'deki derin öğrenme algoritmasında yapılması gereken konvolüsyon işlemini belirli filtrelerle yapmaya yarayan katmandır. Burada 1, 2 ve 3 boyutlu konvolüsyon için kodlamalar bulunur. Bu kodlama işlemleri sayesinde eldeki verinin yükseklik ve genişlik bilgileri, örnek alıcı alan boyutu ile optimize edilir ve giriş hacmine nöronun kaç bağlantısı olduğu tespit edilir. Görüntü matrisler halinde işlenir. Bir sonraki özellik, eşleme değerleri giriş verisi f , konvolüsyon işlemine tabi tutulan Kernel h ve sonuç matrisinin satır ve sütun değerleri sırasıyla m ve n olacak şekilde aşağıdaki denklem ile hesaplanır. [8]

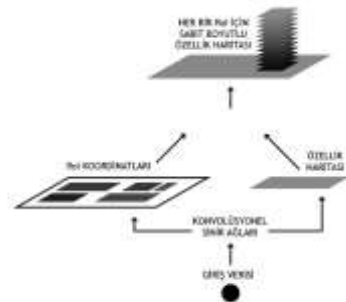
$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \quad (1)$$

ReLU (Rectified Linear Unit) Fonksiyonu

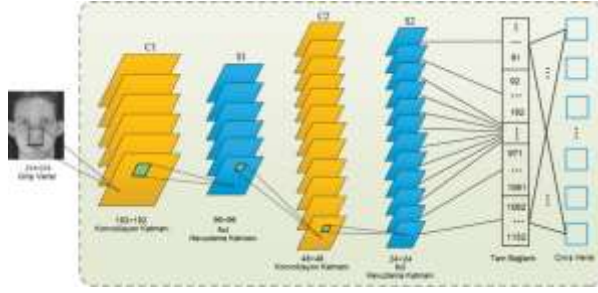
Bu fonksiyon, yapay zekâ için aktivasyon katmanları bölümünün bir dalıdır. ReLU fonksiyonunun en büyük özelliği negatif değerli nöronları aktive etmeyip diğer fonksiyonlara göre daha verimli ve hızlı çalışmasıdır. Konvolüsyon katmanı ve RoI havuzlama katmanı arasında geçiş işlemini gerçekleştiren ara bölümdür [9].

RoI Havuzlama Katmanı

RoI Havuzlama katmanı, havuz boyutunun girdi boyutuna bağlı olduğu bir tür maksimum havuzlamadır. Bu katman, çıktıların her zaman aynı boyutta olmasını sağlar. [8] Bu koşul sağlanmadığı takdirde bir görsel üzerinde farklı boyutlarda birden fazla etiketle karşılaşılabılır. Şekil 1'de roI katmanı ile nesne algılama düzeni görülmektedir.



Şekil 1: RoI katmanı ile nesne algılama düzeni. [10]



Şekil 2: Konvolüsyonel sinir ağlarındaki aşamalar [11].

Bu işlemlerin tümü, CNN algoritmasında arka planda yapılmakta ve elde edilen yüz tanıma verileri ile derin öğrenme sistemi modellemeye entegre edilmektedir. İşlemlerin tek bir aşama olarak gösterildiği Şekil 2’de verilmektedir.

Faster R-CNN

Faster R-CNN mimarisi, R-CNN ve Fast R-CNN’in birlikte çalıştırılarak, geliştirilmesi ile elde edilmiştir. R-CNN’de bir görselden 2000 adet bölge çıkarılır ve IoU (Intersection Over Union) değeri baz alınarak 0,5’ten düşük değerli bölgeler bastırılır. Fast R-CNN’de tek bir bölge oluşturularak R-CNN’de kullanılan CNN, SVM ve Regressor parametreleri birleştirilerek performans artışı sağlanır. Faster R-CNN’de ise RPN (Regional Proposal Network) ve Fast R-CNN birlikte kullanılarak en yüksek performans veren mimari oluşturulmuş olur. Nitekim aynı görsel, R-CNN’de 84 saatte eğitilip 47 saniye tahmin süresine ulaşılırken, bu değere Fast R-CNN’de 8,75 saat eğitimde, 2,3 saniye tahmin süresine ulaşılmıştır. Faster R-CNN, Fast R-CNN ile aynı sürede eğitilmesine rağmen tahmin süresinin 0,3 saniyeye kadar düştüğü gözlemlenmiştir.

Performans Analizi ve Hata Senaryoları

Python programlama dilinin 3.6 versiyonu, Tensorflow açık kaynak kodlu modelleme yapısının 1.15 (GPU) versiyonundan yararlanılarak Anaconda 3 Prompt ile kullanılmıştır. Tüm kodlar Anaconda Prompt üzerinden yazılmış ve derin öğrenme bu program sayesinde gerçekleştirilmiştir.

Öncelikle veri setinde kullanılan maskeli ve maskesiz yüz örnekleri için [1]’deki çalışmanın veri setinden yararlanılmıştır. Burada bulunan 10000’in üzerindeki görseller LabelImg programı ile etiketlenmiş ve derin öğrenme modeline entegrasyonu sağlanmıştır. Fakat görsellerin kalitelerinin düşük olması ve gerçek zamanlı olarak kullanılacak olması dolayısıyla, istenilen performans seviyesine 200000 adıma rağmen ulaşamamıştır. Bu senaryo için rastlanan hatalı örnekler Şekil 3’te gösterilmiştir.



Şekil 3: Hata Senaryosu Örnekleri.

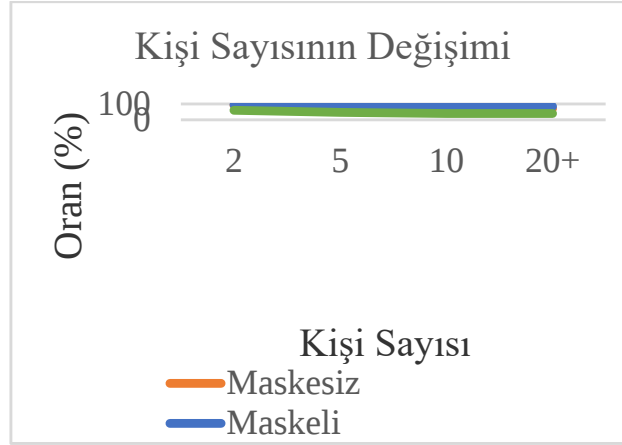
Bu hatalı durumlar çalışmanın sonucunu olumsuz etkileyebileceğinden, kullanılan veri seti değiştirilmiş, daha yüksek kaliteli 5335 adet görselin bulunduğu “Mask Face Lite Dataset” adlı veri seti kullanılmıştır [12]. Bunlara ilave olarak 93 adet yeni görsel eklenmiş ve yukarıdaki hatalı senaryolar, maskesiz yüz olarak etiketlenip 36038 adımda öğretilmiştir. Öte yandan 5 farklı kişinin 100’er adet fotoğrafı isimleri ile etiketlenmiş ve tanıma işlemi için de bu veriler kullanılmıştır. Belirli parametrelere göre sistemin doğruluk oranlarında değişime rastlanmıştır. Bunlar başlıklar halinde aşağıda sunulmaktadır.

Kişi Sayısının Değiştirdiği Senaryolar

Kişi sayısı 5'ten az iken CPU kullanımının %36-%45 arasında değiştiği, RAM kullanımının 1.4 GB-1.7 GB arasında olduğu, kişi sayısının 15'ten fazla iken CPU kullanımının %52-%60 seviyelerine kadar çıktığı, RAM kullanımının ise 1.8 GB-2.1 GB arasında değiştiği gözlemlenmiştir. Doğruluk oranında ise CPU ve RAM kullanımı ile orantılı bir değişim saptanmamıştır. Kişi sayısı artışına bağlı olarak, kişilerin kameraya olan uzaklığı ve duruş açılarına göre algılama başarımları değişkenlik gözlenmekte ve maskesiz yüzlerde ortalama %83,75, maskeli yüzlerde %90 ve tanınan maskesiz yüzlerde ise ortalama %48,5 oranında doğruluk sağlanmaktadır. Kişi sayısının değişimi ile uygulanan senaryolar ve bu senaryolar ile elde edilen doğruluk oranları sırası ile Şekil 4 ve Şekil 5'te verilmektedir.



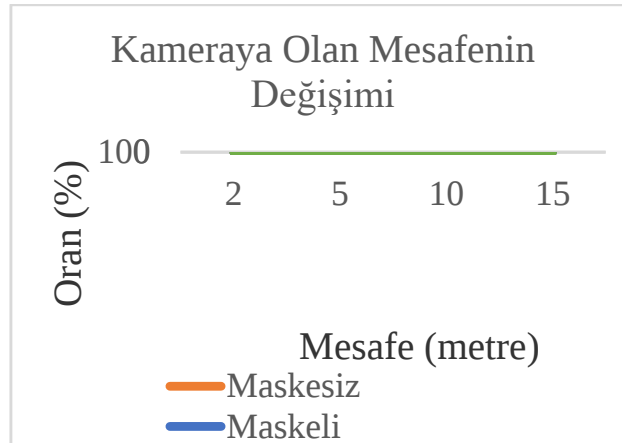
Şekil 4: Kişi Sayısının Değişimi Senaryoları.



Şekil 5: Kişi sayısının değiştiği doğruluk oranları.

Kameraya Olan Mesafenin Değiştiği Senaryolar

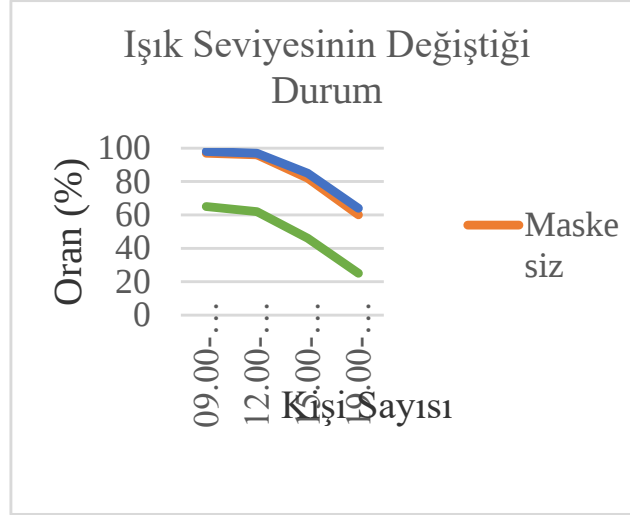
Kullanılan kameradan, 2 m'den fazla uzaklıkta olan insanların bulunduğu senaryolarda, doğruluk oranının %7 oranında azaldığı gözlenmektedir. Mesafe 10 m'nin de üzerine çıktığında ise kameranın algıladığı yüz piksellerinin sayısı da azaldığından sistem doğruluğu olumsuz etkilenmektedir ve algılama başarımındaki (doğruluk oranı) azalma oranının, Şekil 6'da görüldüğü üzere %13'ü geçtiği gözlenmektedir.



Şekil 6: Kamera mesafesinin doğruluk oranına yansması.

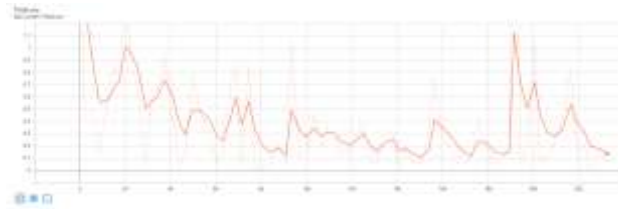
Ortamdaki Işık Seviyesinin Değiştiği Senaryolar

Ortam ışığının yeterli seviyede sağlanmadığı senaryolar, doğruluk oranının en olumsuz şekilde etkilendiği senaryolardır. Bunun sebebi, kullanılan kameranın diyafram değerinin yüksek olması ve dolayısıyla ışık çekim performansının az olmasıdır. Kullanılan kamera kalitesi artırılarak, bu probleme önlem almak mümkün olsa da kameranın kalitesi ne kadar iyi olursa olsun, performansı, aydınlık ortamda çok daha mükemmel olacaktır. 2 m'den az mesafede bulunan senaryolar göz önünde bulundurulduğunda, akşam 18.00'dan önce maskeli ve maskesiz yüzlerdeki doğruluk oranı %95'in üzerinde iken, 18.00 sonrası doğruluk oranının düşüşe geçtiği ve %60-%65 seviyelerine kadar indiği gözlemlenmiş, tanınan maskesiz yüz verisi için de doğruluk oranının %40'ın altında seyrettiği Şekil 7'de görülmüştür.



Şekil 7: Işık seviyesinin doğruluk oranına etkisi.

Yüz tanıma sınıflarında ise daha fazla veriye ihtiyaç duyulduğu sonucuna varılmıştır. Yüz tanıma sınıflarındaki verinin yetersizliği, Şekil 8'de verildiği üzere, toplam kaybı gösteren değişimin kararsız yapısından anlaşılabılır. Bu değişim, TensorBoard uygulamasının bir parçası olup, sisteme entegre edilmiştir.



Şekil 8: TensorBoard uygulaması ile ulaşılan toplam kayıp.

Sonuçlar

Gerçeklenen çalışmalar ve sonuçları incelendiğinde özellikle sağlık alanında ve korunma amaçlı olarak değerlendirilen yüz algılama uygulamalarında, maskeli yüz tanıma problemlerinin derin öğrenme modelleriyle çözülebileceği görülmüştür. Bu çalışmada Faster RCNN Inception V2 modeli ile yüz tanıma üzerinde durulmuştur. Tensorflow kütüphanesi ile bu modelin sınıflandırılması yapılmış ve yeterli veri seti sağlanan sınıflarda başarılı sonuçlar elde edilmiştir. Maskeli ve maskesiz olarak yüz algılama başarımları optimum koşullarda %95'in üzerinde doğruluk oranına ulaşmıştır. Elde edilen sonuçlarda sunulan yöntemin, uygun veri, doğru aydınlatma, mesafe ve kaliteli kamera sistemi sağlandığı takdirde çok başarılı bir uygulama haline gelebileceği görülmektedir. Bu da uygulamanın kullanıldığı ortama göre, maske kullanımı olmaması durumunda gerekli uyarının yapılabileceği, kullanışlı bir denetleme mekanizması oluşturacaktır.

Kaynaklar

- [1] Talahua, J. S., Buele, J., Calvopiña, P., & Varela-Aldás, J. (2021). Facial recognition system for people with and without face mask in times of the COVID-19 pandemic. *Sustainability*, 13(12), 6900. <https://doi.org/10.3390/su13126900>
- [2] Siradjuddin, I. A., Reynaldi, R., & Muntasa, A. (2021). Faster region-based convolutional neural network for mask face detection. 2021 5th International Conference on Informatics and Computational Sciences (ICICoS). <https://doi.org/10.1109/icicos53627.2021.9651744>
- [3] Abbasi, S., Abdi, H., & Ahmadi, A. (2021). A face-mask detection approach based on Yolo applied for a new collected dataset. 2021 26th International Computer Conference, Computer Society of Iran (CSICC). <https://doi.org/10.1109/csicc52343.2021.9420599>
- [4] Z. Ozkan, E. Bayhan, M. Namdar, and A. Basgumus, "Object Detection and Recognition of Unmanned Aerial Vehicles Using Raspberry Pi Platform," IEEE 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (IEEE ISMSIT 2021), pp.467-472, Ankara, Turkey, Oct. 2021.
- [5] E. Bayhan, C. B. Deligoz, F. Seymen, M. Namdar, and A. Basgumus, "Unimpeded Walking with Deep Learning," IEEE 30th Signal Processing and Communications Applications Conference (IEEE SIU 2022), Safranbolu, Turkey, May 2022.
- [6] Punagin, A. (2021, August 2). Inception V2 and V3 - Inception network versions. GeeksforGeeks. Retrieved September 7, 2022, from <https://www.geeksforgeeks.org/inception-v2-and-v3-inception-network-versions/>
- [7] Correspondent, T. G. (2020, December 4). Understanding the difference between AI, ML, and DL. TechGig. Retrieved September 7, 2022, from <https://content.techgig.com/technology/understanding-the-difference-between-ai-ml-and-dl/articleshow/75493798.cms>
- [8] Skalski, P. (2019, April 14). Gentle dive into math behind Convolutional Neural Networks. Medium. Retrieved September 7, 2022, from <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9/>
- [9] Praharsha, V. (2022, January 2). Relu (rectified linear unit) activation function. OpenGenus IQ: Computing Expertise & Legacy. Retrieved September 7, 2022, from <https://iq.opengenus.org/relu-activation/>
- [10] Grel, T. (2021, April 14). Region of interest pooling explained. deepsense.ai. Retrieved September 7, 2022, from <https://deepsense.ai/region-of-interest-pooling-explained/>
- [11] Xie, Z., Li, J., & Shi, H. (2019). A face recognition method based on CNN. *Journal of Physics: Conference Series*, 1395(1), 012006. <https://doi.org/10.1088/1742-6596/1395/1/012006>
- [12] Kottarathil, P. (2020). Face Mask Lite Dataset. Retrieved September 7, 2022, from <https://www.kaggle.com/datasets/prasoonkottarathil/face-mask-lite-dataset>
- [13] Li, F.-F., Wu, J., & Gao, R. (2021). Convolutional Neural Networks (CNNs / ConvNets). CS231N convolutional neural networks for visual recognition. Retrieved September 7, 2022, from <https://cs231n.github.io/convolutional-networks/>
- [14] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>
- [15] Halawa, L. J., Wibowo, A., & Ernawan, F. (2019). Face recognition using faster R-CNN with inception-V2 architecture for CCTV camera. 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS). <https://doi.org/10.1109/icicos48119.2019.8982383>
- [16] Singh, A. (2020, May 8). Introduction to neural network: Convolutional Neural Network. Analytics Vidhya. Retrieved September 7, 2022, from <http://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/>

- [17] Lu, P., Song, B., & Xu, L. (2020). Human face recognition based on Convolutional Neural Network and augmented dataset. *Systems Science & Control Engineering*, 9(sup2), 29–37. <https://doi.org/10.1080/21642583.2020.1836526>
- [18] Nelli, F. (2018). *Python data analytics: With pandas, NumPy, and Matplotlib*. Apress.