

Bulut üzerinde Tensorflow Javascript ile geliştirilen Öğretilebilir Makine (ÖM) Modelinin Doğruluk Tahmini ve Analizi: Yüz Tanıma Sistemi Uygulaması

Accuracy Prediction and Analysis of Teachable Machine (TM) Model developed with Tensorflow Javascript on the Cloud: Face Recognition System Implementation

Ömer Faruk ASLANDOĞDU¹ 

Hakan AYDIN² 

Ali ÇETİNKAYA³ 

DOI:10.33461/uybisbbd.1106753

Öz

Makale Bilgileri

Makale Türü:
Araştırma Makalesi

Geliş Tarihi:
22.04.2022

Kabul Tarihi:
08.06.2022

©2022 UYBİSBBD
Tüm hakları saklıdır.



Yüz Tanıma Sistemleri (YTS), bir kişinin yüzünü kullanarak kimliğini tanımlamanın veya doğrulamanın bir yolu olarak kullanılan bir biyometrik güvenlik kategorisidir. Literatürde YTS çalışmalarında sıklıkla farklı Yapay Zekâ (YZ) teknik ve yöntemlerinin kullanıldığı görülmektedir. Bu modeller çoğunlukla seçilen programlama dilleri ile kodlamada yapılmak suretiyle oluşturulmakta, eğitilmekte, test edilmekte ve bu işlem adımları en yüksek başarı oranı elde edilinceye kadar tekrarlanmaktadır. Öğretilebilir Makine (Teachable Machine, ÖM), YZ modelleri oluşturmayı hızlı, kolay ve herkes için erişilebilir hale getiren web tabanlı bir araçtır. Bu çalışmada herhangi bir kodlamada yapmadan kullanıcıların kendi derin öğrenme modellerini geliştirmelerine imkân sağlayan ÖM teknolojisini kullanmak suretiyle Tensorflow Javascript teknolojileri ile bulut tabanlı bir YTS sisteminin (YTS-ÖM) geliştirilmesi amaçlanmıştır. Çalışma kapsamında gerçekleştirilen deneylerde YTS-ÖM'nin başarı oranı %99.8 olarak ölçülmüştür. Bu çalışmanın özellikle YTS modellerinin kullanıcılar tarafından geliştirilmesi açısından YZ çalışma alanına katkı sağlayacağı değerlendirilmektedir.

Anahtar Kelimeler: Yapay Zekâ, Bulut Bilişim, Yüz Tanıma Sistemi, Öğretilebilir Makine.

Abstract

Article Info

Paper Type:
Research Paper

Received:
22.04.2022

Accepted:
08.06.2022

©2022 UYBİSBBD
All rights reserved.



Face Recognition Systems (FRS) is a category of biometric security used as a way to identify or verify a person's identity using their face. In the literature, it is seen that different Artificial Intelligence (AI) techniques and methods are frequently used in FRS studies. These models are first created, trained, tested by coding with selected programming languages, and these steps are repeated until the highest success rate is reached. Teachable Machine (TM) is a web-based tool that makes building AI models fast, easy and accessible to everyone. In this study, it is aimed to develop a cloud-based FRS system (FRS-TM) with Tensorflow Javascript technologies by using the TM technology that allows users to develop their own deep learning models without any coding. In the experiments carried out within the scope of the study, the success rate of FRS-TM was measured as 99.8%. It is thought that this study will contribute to the field of AI study, especially in terms of the development of YTS models by users.

Keywords: Artificial Intelligence, Cloud Computing, Face Recognition System, Teaching Machine.

Atf/ to Cite (APA): Aslandoğdu, Ö. F., Aydın, H., ve Çetinkaya, A., (2022). Bulut üzerinde Tensorflow Javascript ile geliştirilen Öğretilebilir Makine (ÖM) Modelinin Doğruluk Tahmini ve Analizi: Yüz Tanıma Sistemi Uygulaması. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 6(1), 66-77

¹ İstanbul Gelişim Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, omeraslandogdu@gmail.com, İstanbul, Türkiye.

² İstanbul Topkapı Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, hakanaydin@topkapi.edu.tr, İstanbul, Türkiye.

³ İstanbul Gelişim Üniversitesi, Teknoloji Transfer Ofisi Uygulama ve Araştırma Merkezi, alcetinkaya@gelisim.edu.tr, İstanbul, Türkiye.

1. GİRİŞ

Bir kişinin yüzünü kullanarak kimliğini tanımlamanın veya doğrulamanın bir yolu olarak kullanılan Yüz Tanıma Sistemleri (YTS) bir biyometrik güvenlik kategorisi olarak sıklıkla fotoğrafları, videoları veya gerçek zamanlı olarak kişileri tanımlamak için kullanılmaktadır. Parmak izlerinin benzersiz olması gibi, her insanın kendine özgü bir yüz izi vardır. YTS'ler ile yüz izleri cinsiyet, yaş veya ırk gibi kategorilere ayrılabilen, veri tabanlarında depolanabilmekte ve diğer yüz izleriyle karşılaştırılabilmektedir. YTS ana faktörleri arasında gözler arasındaki mesafe, göz çukurları derinliği, alından çeneye olan mesafe, elmacık kemiklerinin şekli ve dudaklar, kulaklar ve çenenin hatları bulunur. Amaç, yüzü ayırt edici kılan belirgin noktaları tanımlamaktır. Yüz yakalama işlemi, kişinin yüz özelliklerini baz alarak dijital veriye dönüştürür. Böylelikle yüzün analizi aslında temelde bir matematik formülüne dönüştürülmektedir. Bu sayısal koda yüz izi adı verilmektedir. Yapay Zekâ (YZ) teknolojilerinin de kullanıldığı YTS'ler ile belirli yüz özelliklerinin geometrisi kullanılarak insanların yüz izleri oluşturulmakta, bu maksatla çeşitli YZ modelleri çok sayıda görüntü kullanılarak eğitilmekte, eğitilen modeller test edilmekte ve doğruluk oranları hesaplanmaktadır. YTS teknolojisi, dijital görüntü işleme, nesne tanıma, desen tanıma, kod tanıma ve optik karakter tanıma gibi teknolojiler gibi görüntü işleme alanı kapsamında değerlendirilmektedir. Günümüzde YTS'lerden tarım, tıp, eğitim, sağlık ve güvenlik gibi birçok alanda günlük yaşantımızda faydalanılmaya başlanmıştır. Görüntü işleme teknikleri tıp, askeri sistemler, endüstriyel ve coğrafi sistemler başta olmak üzere birçok alanda yaygın olarak kullanılmaktadır (Eldem ve diğ., 2017). Özellikle içinde bulunduğumuz COVID-19 Pandemi döneminde YTS kullanımı, personelin kart basmasını veya parmak izi okutmasını önleyerek hijyen ve sağlık koşulları açısından olumlu faydalar sağlayabilecektir.

Literatürde YZ çalışmaları kapsamında YTS'ler ile yapılmış pek çok çalışma olduğu görülmektedir. Söz konusu çalışmalarda sıklıkla yüzleri ayırt etmek için Derin Öğrenme (DL), Makine Öğrenmesi (ML) ve YSA (Yapay Sinir Ağları) gibi farklı YZ teknik ve yöntemlerinin kullanıldığı görülmektedir. Bu bağlamda seçilen programlama dilleri ile kodlamada yapılmak suretiyle aktivasyon fonksiyonu, iterasyon sayısı gibi parametreler dikkate alınarak YTS modelleri oluşturulmakta, eğitilmekte ve test edilmekte, bu işlem adımları en yüksek başarı oranı elde edilinceye kadar tekrarlanmaktadır. ÖM teknolojisi, ilk versiyonu 2017 yılında yayınlanmış olan ve en kısa tanımı ile makine öğrenimi modelleri oluşturmayı hızlı, kolay ve herkes için erişilebilir olan web tabanlı bir araçtır. ÖM, kullanıcının web tarayıcısında yaptığı modelleri eğitmek ve çalıştırmak için Javascript'te ML için kullanılan TensorFlow.js kitaplığını kullanır. Bu teknoloji ile kullanıcılar herhangi bir kodlamada yapmadan kendi DL modellerini geliştirebilirler. Bu çalışmada herhangi bir kodlamada yapmadan kullanıcıların kendi derin öğrenme modelini geliştirmeye imkân sağlayan Öğretilebilir Makine (ÖM) kullanmak suretiyle Tensorflow Javascript teknolojileri ile bulut tabanlı bir YTS sisteminin (YTS-ÖM) geliştirilmesi amaçlanmıştır. Bu amaç doğrultusunda geliştirilen sistem bulut tabanlı Web uygulamasına bağlanabilmekte, bir internet bağlantısı ve tarayıcısına sahip bir cihaz üzerinden yine bulut tabanlı olarak çalıştırılabilmektedir. YTS-ÖM sisteminde üretilen veriler yoklama sistemine kaydedilmek suretiyle öğrenci yoklama kaydının tutulması amacıyla kullanılmaktadır. Çalışmada ÖM teknolojisi yanında ayrıca TensorFlow.js ile Keras vtfjs-layers kütüphaneleri, Python Django Web Geliştirme Arayüzü ve PostgreSQL veri tabanı teknolojileri de kullanılmıştır. Çalışma kapsamında gerçekleştirilen deneylerde ÖM ile oluşturulan YZ modelinin başarı oranı %99.8 olarak ölçülmüştür. Bu çalışmanın literatüre katkıları şu şekilde ifade edilebilir:

- YTS-ÖM sisteminin geliştirilmesinde, herhangi bir kodlamada yapmadan kullanıcıların kendi derin öğrenme modelini geliştirmeye imkân sağlayan ÖM teknolojisi kullanılmıştır.

- Önerilen sistemin geliştirilmesinde ÖM teknolojisi yanında ayrıca TensorFlow.js, Keras vtfjs-layers, Python Django Web Geliştirme Arayüzü ve PostgreSQL veri tabanı teknolojileri de kullanılmıştır. YTS-ÖM'nin ara yüz tasarımı Python Django kütüphanesi ile Rest Framework kullanılarak geliştirilmiştir. Django kütüphanesinin sağladığı API yapısı ile istemciden gelen isteklerin sunucuda işlenmesi hedeflenmiştir. Özelleştirilebilir yapısı için Django kütüphanesi yerine

Flask teknolojisinin de çalışmada kullanılması değerlendirilmiş, ancak ilave kullanıcı isterlerinin yapılandırılması göz önüne alınarak çalışma çatısının esnek tutulması esas alınmıştır.

- Bulut Bilişim (BB) teknolojisinin bilgi güvenliği gibi birtakım dezavantajları yanında geniş yelpazede sunduğu pek çok avantaj ortadadır. Bu bağlamda YTS-ÖM sistemi bulut tabanlı bir sistemdir.

Araştırmada elde edilen sonuçlar, kullanıcıların kendi derin öğrenme modellerini kodlama yapmadan geliştirmelerine imkân sağlayan ÖM teknolojisi kullanılarak geliştirilen YTS modelinin yüzleri tespit etmedeki etkinliğini göstermektedir. Bu çalışmanın özellikle güvenlik endişesi nedeniyle birçok araştırmacıyı ve kullanıcıyı cezbeden IP sistemlerinin kullanıcılar tarafından geliştirilmesi açısından YZ çalışma alanına katkı sağlayacağı değerlendirilmektedir.

Çalışmanın 2'inci bölümünde literatür taraması ile elde sonuçlara yer verilmiştir. 3'üncü bölümde çalışmaya ilişkin materyal ve metot anlatılmaktadır. 4'üncü bölümde çalışmada gerçekleştirilen deneysel sonuçlar açıklanmaktadır. Son olarak, 6'ıncı bölümde çalışmaya ilişkin sonuç ve değerlendirmelere yer verilmiştir.

2. İLGİLİ ÇALIŞMALAR

Sütçüler (2006) tarafından gerçekleştirilen çalışmada yüz tanıma için yüz piksel renklerinin kullanımı üzerine bir araştırma gerçekleştirilmiştir. Bu maksatla beş yüz binin üzerinde yüz piksel renkleri sınıflandırılarak ten rengi tespitiyle yüzün bulunduğu konum ve algılaması sağlanmıştır. Araştırmada yapılan testlerde 90% başarı oranı elde edilmiştir. Ayrıca yüz tanıma sisteminde yüzü algılanacak insanların sistemde beşten fazla fotoğrafının bulunması halinde başarı oranının %91 oranına çıktığı da çalışmada belirtilmektedir.

Gökberk (2006) tarafından gerçekleştirilen çalışmada üç boyutlu yüz algılama yapan bir ürün tasarlanmıştır. Sistem çakıştırma, betimleme, öznitelik elemine etme, karar bütünleştirme adımları ile çalışmaktadır. Yazıcı (2008) tarafından gerçekleştirilen çalışmada; Alışveriş Merkezlerinde (AVM) kullanılması için müşterilerin yüzlerini algılayan bir sistem tasarlanmıştır. Önerilen sistemde video üzerinden çalışan sistem müşterilerin memnun olup olmadığını algılamaktadır. Sistemde ayrıca müşterilerin kaç gün sonra tekrar geldiği, son geldiği gün, sesli mesaj ile müşteriye özel karşılama gibi modüller de bulunmaktadır. (Tırkaz, 2008) tarafından gerçekleştirilen çalışmada önerilen AGM'nin (Aktif Görünüm Modeli) avantajları ve oluşturulabilecek uzantılar araştırılmıştır. Bu bilgiler doğrultusunda modelin görüntülerde gelen objeleri tespit edebildiği ortaya konmuştur.

Hazar (2008) tarafından gerçekleştirilen çalışmada yüz tanıma işlemi için yüz ve yüzdeki organların önemi ortaya konmuştur. Topkaya (2008) tarafından gerçekleştirilen çalışmada video görüntüleri kullanılarak bir yüz tanıma sistemi geliştirilmiştir. Önerilen sistemde eğitilen model videolardaki yüzleri tanımaya çalışmaktadır. Manav (2010) tarafından gerçekleştirilen çalışmada Matlab kullanılarak biyometrik göz tarama sistemi geliştirilmiştir. Göz fotoğrafları sadece gözün çemberi dikkate alınarak oluşturulmuş ve diğer veriler dikkate alınmamıştır. Önerilen sistem insan hareketlerini algılayarak göz tanıma sistemini aktif hale getirmektedir. Sistemde göz çemberi tespit edilmekte ve fotoğrafı veri tabanına kayıt edilmektedir. Çalışmada ayrıca önerilen sistem ile iki yüzden fazla göz örneği oluşturulmuştur.

Eleyan (2010) tarafından gerçekleştirilen çalışmada yüz tanıma için dalgacık dönüşümü sistemleri ile öznitelik çıkarım çalışması gerçekleştirilmiştir. Dalgacık dönüşümü olarak "Gabor" ve "Çift Ağaç" yapıları kullanılmıştır. Çalışmada dört farklı yaklaşımdan bahsedilmektedir. İlk iki yaklaşım dalgacık dönüşüm yöntemleri ile oluşturulan vektörleri sinir ağının eğitiminde kullanılmaktadır. Diğer iki yöntem ise bu ilk iki yöntemin artarda eklenmesi ile oluşturulmuştur. Oluşturulan sistemler arası performans testleri için simülasyonlar ve performans ölçümleri yapılmıştır. Yıldırım (2011) tarafından gerçekleştirilen çalışmada yüz tanıma için "En Küçük Kareler Metodu" ve "Ridge Regresyon Metodu" hataları baz alınmıştır. SPSS ve Matlab kullanılarak C++ dili ile geliştirilen sistemde yüzün belirli bölgelerinden alınan verilerle işlem yapılmaktadır.

Filiz (2012) tarafından gerçekleştirilen çalışmada YTS'lerin geliştirilmesinde literatürde kullanılan farklı teknik ve yöntemler incelenmiş ve çalışmada YTS algoritması olarak "SIFT Algoritması" kullanılmıştır. Bu algoritma farklı veri setlerinde farklı yöntemlerle test edilmiştir. Ayhan (2013) tarafından gerçekleştirilen çalışmada yüz tanıma için fotoğrafın nasıl seçilmesi gerektiği üzerine araştırma yapılmıştır. Başlangıç noktası için insan gözü seçilmiş ve yüzün etrafı bu durum dikkate alınarak taranmıştır. Geliştirilen çalışmanın test işlemleri için "SFace verisetinde" yer alan fotoğraflar kullanılmıştır.

Holat (2014) tarafından gerçekleştirilen çalışmada YTS kullanılarak bir kimlik algılama sistemi tasarlanmıştır. Çalışmada Adaboost yöntemi tercih edilmiştir. Çalışma sonuçlarında, Histogram eşleme ve bazı filtreler kullanıldığında %6'ya kadar başarı artışı gözlemlendiği belirtilmiştir. Aslan (2016) tarafından gerçekleştirilen çalışmada sabit kamera kullanan bir YTS tasarlanmıştır. Çalışmada sabit kamera açısının getirdiği kısıtlar, kameraya olan yakınlık-uzaklık, duruş farklılıkları gibi esas alınmıştır. Saeed (2016) tarafından gerçekleştirilen çalışmada yüz algılama ve tanıma için literatürde kullanılan farklı teknik ve yöntemler araştırılmıştır. Al Bazzaz (2017) tarafından gerçekleştirilen YTS çalışmasında Nesnelerin interneti (IoT) cihazı olarak "Arduino Uno" kullanılmıştır. Bu cihaz ile nesne tanıma ve nesne takibi yapılmıştır.

Bilgiç ve diğ. (2017) tarafından yapılan çalışmada, farklı yüz ifadeleri ve aydınlatma koşullarına sahip 40 kişiye ait Kinect RGB görüntü veri seti dört kat çapraz doğrulama yöntemi ile test edilmiştir. Aghdam (2018) tarafından gerçekleştirilen çalışmada fotoğrafların kalitesi, ışık, pozlama gibi farklılıkların yüz tanımadaki performansı ve başarımı üzerine bir araştırma yapılmıştır. Veri seti olarak ise "VGGFace2", "ICB-RW" ve "SCFace" kullanılmıştır. Rashid (2018) tarafından gerçekleştirilen çalışmada dünyada artan terör ve suç oranlarına dikkat çekilerek YTS'lerin önemine vurgu yapılmıştır. Çalışmada ayrıca Microsoft Azure veri tabanına görüntüleri kaydederek eşleşen özelliklere göre kapının kilidini açıp kapatabilen bir mekanizma da tasarlanmıştır. Çambay (2019) tarafından gerçekleştirilen çalışmada farklı işlemciler ve grafik kartları ile nesne algılama ve nesne tanıma işlemleri yapılarak FPGA'ların düşük karar düzeyinde az güçle daha iyi sonuçlar verdiği gözlemlenmiştir. İlk olarak PYNQ FPGA kartına OpenCV ile sınıflandırıcı olarak Haar-Kaskat kurulmuş ve ikinci olarak MNIST el yazısı tanıma CIFAR nesne tanıma veri setleri üzerinde çalışılarak yazılım ve donanım kaynaklı tanıma hızları karşılaştırılmıştır.

Fawzy (2019) tarafından gerçekleştirilen çalışmada araçlar için panel üzerinde gelişmiş sürücü destek sistemi tasarlanmıştır. Çalışmada "OpenGL grafik arayüzü" kullanılarak panel üzerinde işlenmiş görüntüler entegre bir sistem olarak sunulmuştur. Oluşturulan sistemin başarı oranı 79% olarak ölçülmüştür. Noori (2019) tarafından gerçekleştirilen çalışmada "Temel Bileşen Analizi" yöntemiyle yüz algılaması yapılmıştır. Çalışmada yüz algılamada ışık, poz, uzaklık gibi durumlar dikkate alınarak bu durumların her biri için resimler çekilmiştir. Otuz kişi için üç bin altı yüz fotoğraf çekilerek özgün bir veri seti oluşturulmuştur. Çalışmada ORB yöntemi ile 86,29%'luk başarı oranına ulaşıldığı belirtilmiştir.

Akın (2019) tarafından gerçekleştirilen çalışmada DL ile geliştirilen YTS modeli anlatılmaktadır. Çalışmada cinsiyet algılamada 98,25% ve yüz tanımda ise 96,25% başarı oranı elde edildiği belirtilmektedir. Karaman (2019) tarafından gerçekleştirilen çalışmada yüz tanıma sistemlerinin adımları araştırılmıştır. Sertkaya (2019) tarafından gerçekleştirilen çalışmada yüz tanıma için oluşturulacak algoritmanın ve veri setindeki fotoğrafların seçilmesi, kalite ölçümü ve iyileştirilmesi üzerine bir çalışma yapılmıştır. Saray Çetinkaya ve Sertbaş (2022) tarafından yapılan çalışmada CPU (Central Processing Unit - Merkezi işlem birimi) ve GPU (Graphics Processing Unit - Grafik İşlemci Ünitesi) mimarileri görüntü işleme ve zaman bağlı algoritmalar üzerinde üç farklı veri seti üzerinde CNN (Convolutional Neural Network - Evrişimli Sinir Ağı), RNN (Recurrent Neural Network - Yinelemeli Sinir Ağı) ve LSTM (Long Short-Term Memory - Uzun Kısa Süreli Bellek) derin öğrenme algoritmaları uygulanarak karşılaştırılmıştır. Ayata ve Çavuş (2022) tarafından yapılan çalışmada Makine Öğrenmesi (ML) tekniklerinden; ESA (Evrişimsel Sinir Ağları), YGH-DVM (Yönelimli Gradyan Histogramı- Destek Vektör Makineleri) ve DSA (Derin Sinir Ağları)

yöntemleri kullanılarak FEI, CelebA ve Aile yüz veri seti üzerinde analizler yapılmıştır. Bu testler sonucunda ESA yöntemi ile; FEI Veri Setinde %98.86, CelebA Veri Setinde %99.89, Aile Veri Setinde ise %100, YGH-DVM yöntemi ile; FEI Veri Setinde %97.71, CelebA Veri Setinde %97.75, Aile Veri Setinde ise %95.67, DSA yöntemi ile de; FEI Veri Setinde %97.61, CelebA Veri Setinde %95.79, Aile Veri Setinde ise %91.83 başarılı yüz bulma tahminine ulaşılmıştır.

3. MATERYAL VE METOT

3.1 Öğretilebilir Makine (Teachable Machine, ÖM) Teknolojisi

Bu bölümde çalışmamız Öğretilebilir Makine (ÖM) üzerinde gerçekleştirildiğinden dolayı bu teknoloji hakkında bilgi verilmiştir. ÖM teknolojisi insanların kodlama yapmadan, web kameralarını, görüntülerini veya seslerini kullanarak kendi ML sınıflandırma modellerini eğitmelerini sağlayan web tabanlı bir arayüzdür (Carney ve diğ., 2020). Görüntüler veya ses örnekleri içindeki kalıpları ve eğilimleri bulmak ve saniyeler içinde basit ve kolay bir sınıflandırma modeli oluşturmak için bir ML tekniği olan transfer öğrenmeyi kullanır. Aktarım öğrenimi ile, bir kullanıcı kendi verilerini ekleyebilir ve büyük bir veri kümesinden belirli bir etki alanını öğrenen önceden eğitilmiş bir temel modelin üzerine bir modeli yeniden eğitebilir. Örneğin, Teachable Machine'in görüntüler bölümünün temel modeli (mobilenet) başlangıçta 1000 sınıfı (köpek, telefon, yatak, trombon vb.) tanımak üzere eğitilmiştir. Mobilenet'in bu sınıfları tanımak için kullandığı temel özellikler, kullanıcının tanımladığı yeni sınıfları tanımlamak için kullanılabilir. Bu karmaşıklık, kullanışlı ve doğru modeller oluşturmak için daha az veriye ve eğitim süresine ihtiyaç duyarak fayda sağlayan kullanıcılardan gizlenmiştir. Kullanıcılardan iyi eğitim örnekleri vermelerini istemek, teknolojinin gerçekleştirmeyi amaçladığı becerilere benzer beceriler gerektirdiğinden, erişilebilirlik için ÖM teknolojisi benzersiz bir zorluk teşkil eder (Kacorri, 2017). Örneğin, görme engelli kullanıcılara uygulamalarına kendi adlarına "görmeyi" öğretmelerine yardımcı olan uygulamaları nasıl geliştirebilir. ML ve DL modellerinin verileri öğrenmesi için karmaşık matematiksel hesaplar ve formüllere ihtiyaç vardır. Ancak ÖM teknolojisi ile bir kullanıcı matematiksel ve YZ hesaplamaları bilgisi olmadan tarayıcısında ML modelleri oluşturabilir. ÖM teknolojisinde kullanıcıların bir web uygulaması veya bir hesap oluşturmasına gerek yoktur. ÖM teknolojisi ile kullanıcıların web kameralarından veya görüntü dosyalarından çekilmiş görüntülerle ve mikrofonlardaki seslerle YZ modelleri eğitilebilmektedir. Hatta ÖM ile oluşturulmuş olan modeller Google Sunucuları aracılığıyla kullanıcılar tarafından da Teachable Makineleri ile de yayınlanabilmektedir. ÖM ile görüntüler, sesler veya pozlar için kullanıcılar kendi görüntü dosyalarını yükleyebilir veya bir mikrofon veya web kamerasıyla canlı olarak yakalanabilir. Bu örnekler kullanıcıların cihazlarında kalır ve kullanıcılar projelerini Google Drive'a kaydetmeyi seçmediği sürece bilgisayarlarından alınmaz, yani İnternet ortamında herhangi bir ortama kayıt olmaz. Teachable Machine uzantısı, kullanıcılara bir makine öğrenimi modelinin eğitiminin ve kullanımının nasıl çalıştığını öğretmeyi amaçlayan bir tarayıcı uygulaması olan Google'ın Teachable Machine'i temel alır (Zhu, 2019). ÖM teknolojisi kullanmak suretiyle;

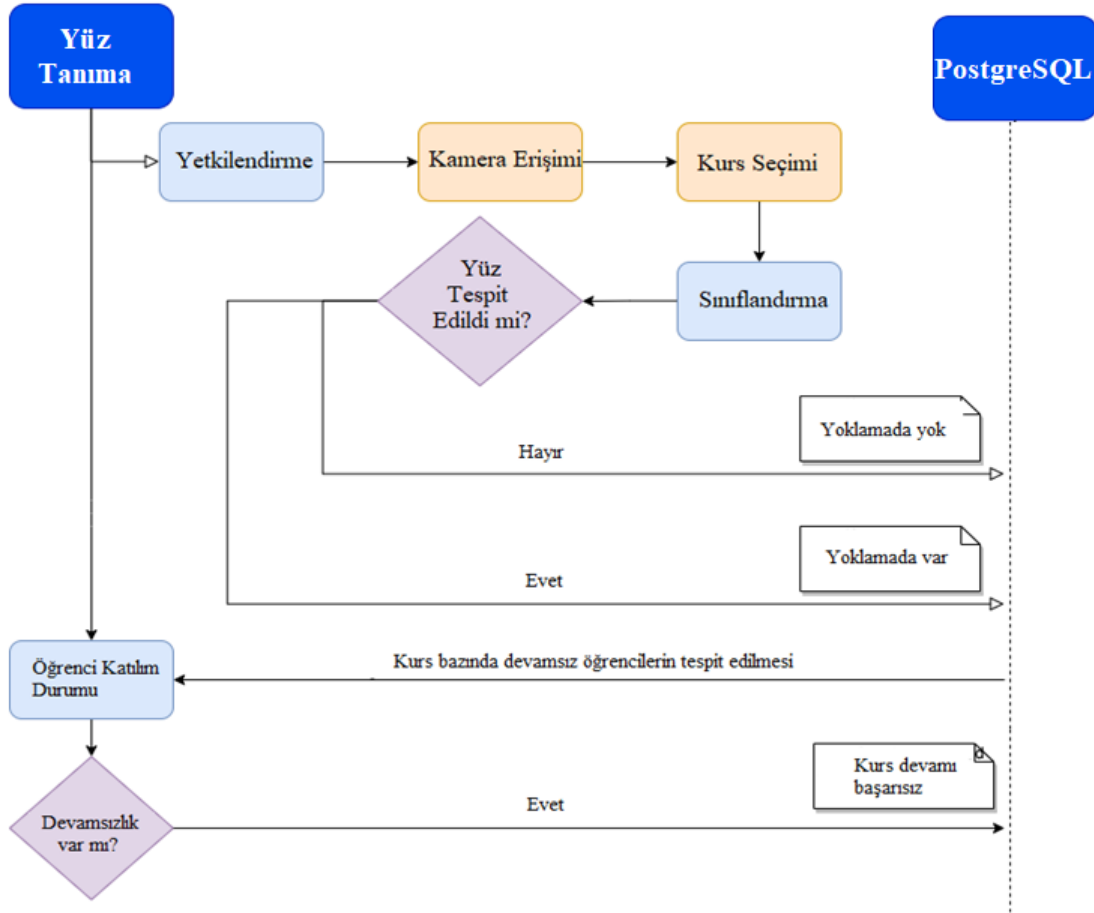
- Kullanıcılar örneklerini, modelin öğrenmesini istedikleri sınıf veya kategoriler halinde gruplayabilirler.
- Modellerini tek bir buton ile eğitip, ardından yeni örnekleri doğru şekilde sınıflandırılıp sınıflandırılmadığını test edebilirler.
- Modellerini siteler, uygulamalar gibi birçok alan için ücretsiz bir şekilde indirebilirler.

Bu hizmet aslında Google Firması tarafından verilmekte olan bir hizmettir. Bu ortam topluluk destekli geliştirilen açık kaynak kodlu görüntü sınıflandırma teknikleri için makine öğrenme modelleri üretebilmeyi sağlayan bir platformdur. Bu sistemde bir yüz sınıflandırması için saniyeler içinde kameradan gelen görüntünün birçok resmini çekilmekte ve sisteme otomatik olarak tanıtılmaktadır. Google sunucularında eğitilen YTS modeli, proje üzerinde kullanmak için istendiği takdirde kullanıcılar tarafından kişisel bilgisayara alınabilmekte veya kaynak kodu indirilmek suretiyle tüm işlemler kullanıcı tarafından yapılabilmektedir. YTS modelinin eğitimi için hızlı ve

tutarlı veriler verdiği için hazır modeller kullanmak yerine ÖM ile eğitilmiş özelleştirilmiş modeller kullanmak yüz tanıma konusunda hız ve kesinlik sağlamaktadır (Google, 2017).

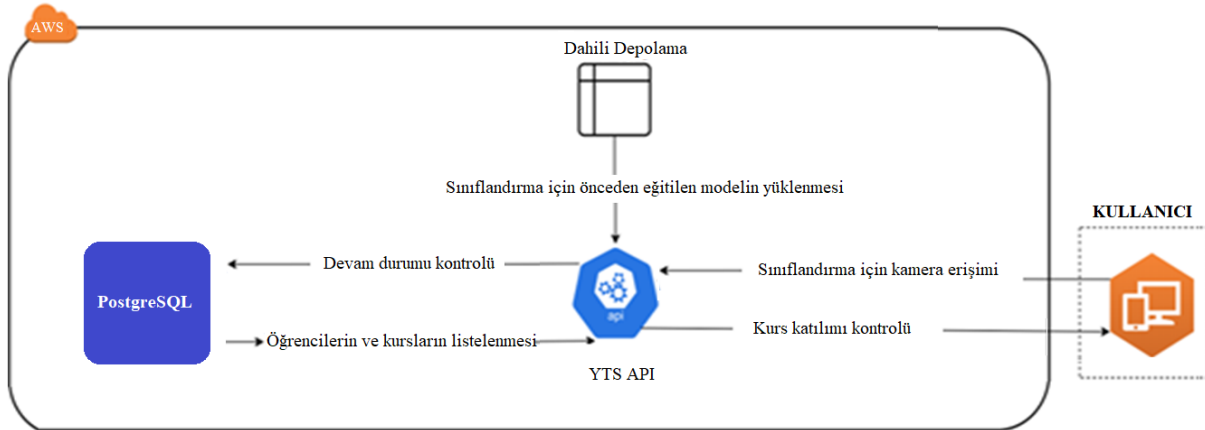
3.2 Çalışmada Önerilen Sistem

Çalışmada önerilen sistemin Akış Şeması Şekil 1'de sunulmuştur.



Şekil 1: YTS-ÖM Akış Diyagramı.

Söz konusu akış diyagramına göre kullanıcı web tarayıcısı üzerinden istek gönderdiğinde “Yüz Tanıma ve Katılım” API’si tarafından yakalanmaktadır. Daha sonra sunucu içindeki depolama alanından YTS modeli yüklenmektedir. Son olarak ise yüklenen model ile gelen görüntü üzerinde sınıflandırma işlemi yapılarak öğrencilerin yüz tanıma işlemleri tamamlanmaktadır. Bu bağlamda sistemin çalışma adımları Şekil 2’de sunulmuştur.



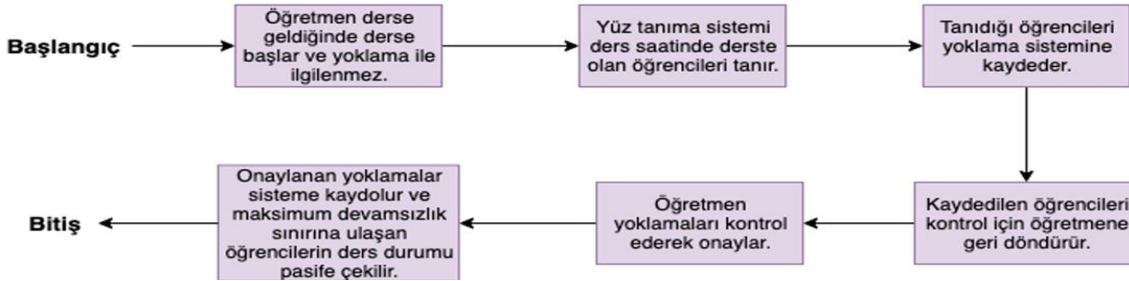
Şekil 2: Sistemin çalışma adımları.

Söz konusu çalışma adımları incelendiğinde sistemin dört farklı çalışma adımının olduğu görülmektedir. Bu adımlardan 1'inci adım, yüz algılama işlemlerini kapsamaktadır. Bu adımda kamera yüz görüntüsünü algılar ve bulur. 2'nci adım yüz analizi işlemlerini kapsamaktadır. Bu adımda yüzün görüntüsü kaydedilip analiz edilir. Bu adımda YTS-ÖM yüzün geometrisini okumaktadır. 3'üncü adım görüntüyü veriye çevirme işlemlerini kapsamaktadır. 4'üncü ve son adım eşleştirme işlemlerini kapsamaktadır. YTS-ÖM sisteminin çalışma adımlarında belirtildiği şekilde sistem tarafından yüz bilgileri tanınan öğrenciler, kullanıcı tarafından gönderilen sınıf bilgisine göre bu çalışma kapsamında oluşturulmuş olan veri tabanına kaydedilmektedir.

Araştırmada önerilen YTS-ÖM sisteminin geliştirilmesine ilişkin yapılan analiz çalışmaları neticesinde oluşturulan diyagramlar **Şekil 3** ve **Şekil 4**'te gösterilmektedir. Söz konusu diyagramlar kullanıcıların ürün tasarlanmadan önceki ve sonraki davranışlarını açıklamaktadır. Proje kullanıcı deneyimini bir üst seviyeye taşımayı amaçlamaktadır. Akademik personelin üzerindeki ders devam durumunu kontrol yükünü alarak akademik araştırmalara da fayda sağlamayı hedeflemektedir. Öğrencilerin derse odaklanmasını sağlamaktır.



Şekil 3. Eski Kullanıcı deneyim akışı.



Şekil 4. Yeni kullanıcı deneyim akışı.

3.3 Veri Seti

Google ÖM teknolojisi, insanların kodlama yapmadan, web kameralarını, görüntülerini veya seslerini kullanarak kendi ML sınıflandırma modellerini eğitmelerini sağlayan web tabanlı bir ara yüzdür. Görüntüler veya ses örnekleri içindeki kalıpları ve eğilimleri bulmak ve saniyeler içinde basit ve kolay bir sınıflandırma modeli oluşturmak için bir ML tekniği olan Aktarım Öğrenmeyi (Transfer Learning, AÖ) kullanır. AÖ tekniği ile, bir kullanıcı kendi verilerini ekleyebilir ve büyük bir veri kümesinden belirli bir etki alanını öğrenen önceden eğitilmiş bir temel modelin üzerine bir modeli yeniden eğitebilir. Bu bağlamda veri seti olarak sistem tarafından daha önce sınıfları tanımak için kullandığı temel veri seti kullanılmış, ancak sistem bu çalışma kapsamında oluşturulan olan ve 10 öğrenciye ait olan yüz görüntüleri test edilmiş ve sistemin başarı oranı hesaplanmıştır. Bu bağlamda örneğin, ÖM'nin görüntüler bölümünün temel modeli (mobilenet) orijinal olarak 1000 sınıfı (köpek, telefon, yatak, trombon vb.) tanımak üzere eğitilmiştir. Mobilenet'in bu sınıfları tanımak için kullandığı temel özellikler, kullanıcının tanımladığı yeni sınıfları tanımlamak için kullanılabilir. Bu karmaşıklık, kullanışlı ve doğru modeller oluşturmak için daha az veriye ve eğitim süresine ihtiyaç

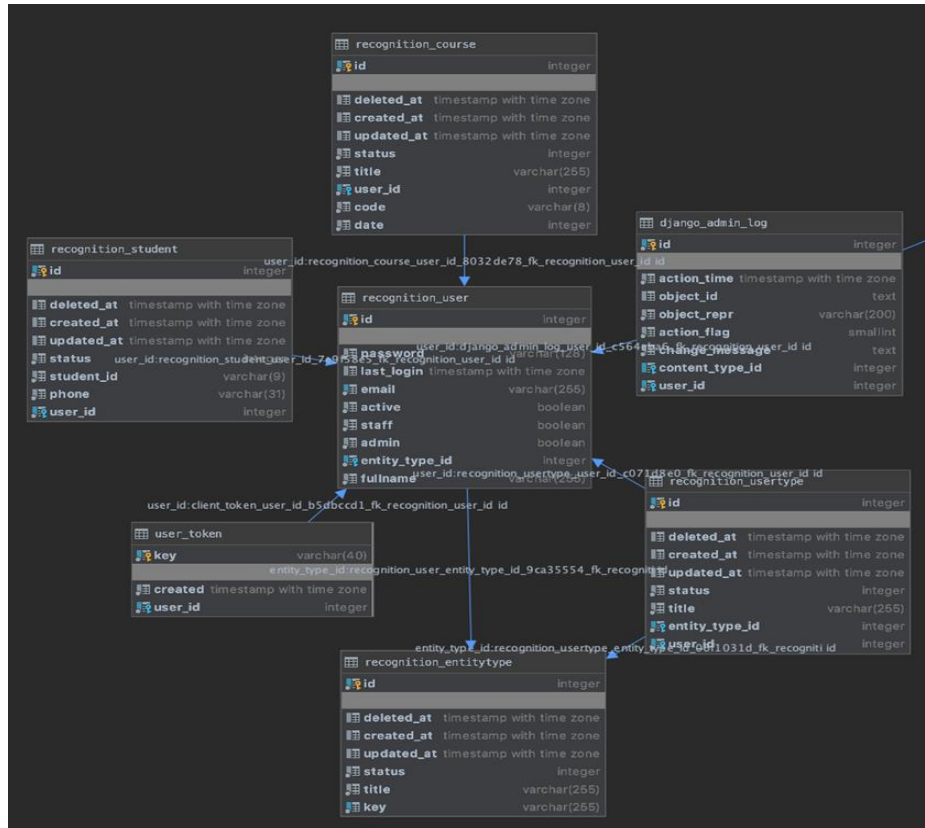
duyarak fayda sağlayan kullanıcılardan gizlenmiştir. ÖM, kullanıcıların modellerini kendi verileriyle, bu veriler bilgisayarlarından ayrılmadan cihazlarındaki tarayıcıda eğitmektedir. Bu özelliği ile ÖM veri gizliliğini sağlayarak kullanıcıların deney yaparken kendilerini güvende hissetmelerine yardımcı olmaktadır. Bu durum ayrıca uzak bir makineye veri gönderen ve işleyen sistemlerden daha fazla gizlilik ve sahiplik duygusuna ve büyük dosyaları, veri kümelerini depolama ve kaydetme konusunda endişelenmenize gerek kalmadan makine öğrenimi ile oynamak ve denemeler yapmak için daha esnek ve daha az kalıcı bir yapıya olanak tanımaktadır.

3.4 Programlama Dili

Çalışmada Python programlama dili, TensorFlow.js yanında ayrıca Keras vtfjs-layers kütüphanesi, Python Django Web Geliştirme Arayüzü ve PostgreSQL veritabanı kullanılmıştır. Tensorflow açık kaynak kodlu bir derin öğrenme kütüphanesidir. Bu kütüphaneye ilişkin kodlar herkesin erişimine açıktır. Tensorflow kütüphanesi platformdan bağımsız bir şekilde işlemci veya grafik kartı üzerinden özellikle ML ve DL gibi algoritmalarının çalıştırabilmesine olanak sağlar. Temeli Python dili ile geliştirilen bu framework Java, R, C++, Go, Javascript gibi birçok dili desteklemektedir. Tensorflow kullanıcılara WEB, nesnelerin interneti (IoT), mobil cihazlar için ayrı ayrı kütüphaneler sağlamaktadır. Web üzerinde geliştirme sağlayan kütüphane olan Tensorflow.js istemci taraflı çalışmaktadır. Tensorflow.js ile istemci üzerinde model eğitebilir ve tahminleri gerçekleştirebilir.

3.5 Veritabanı

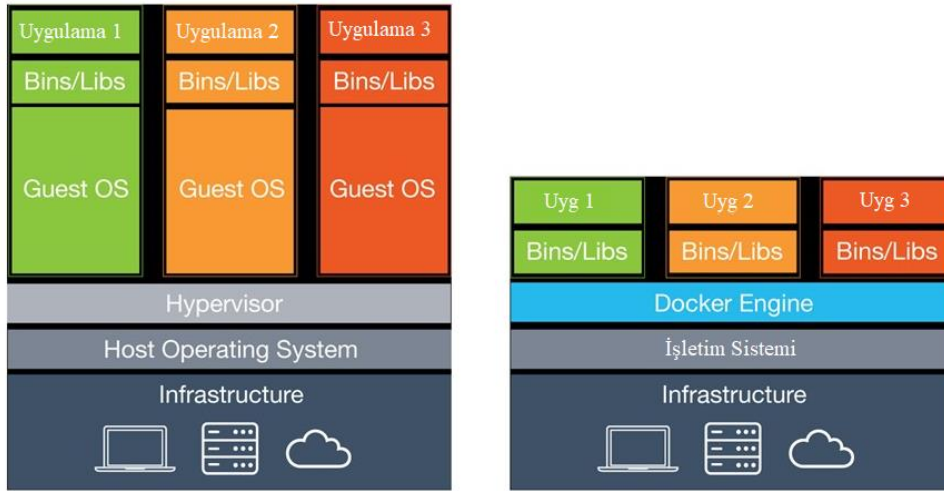
Çalışmada kullandığımız PostgreSQL veritabanı açık kaynak kodlu bir ilişkiyel veri tabanıdır. PostgreSQL veri tabanı bu türü destekleyerek farklı programlama dillerinde yazılmış farklı modüller ile hazırlanmış büyük uygulamaların birbiriyle veri değişimine imkân tanımaktadır. Çalışmanın veritabanı ilk defa bu çalışma kapsamında geliştirilmiştir. Çalışmada geliştirilen veritabanı Şekil 5’de sunulmuştur.



Şekil 5: Veritabanı Tasarımı

3.6 Docker Teknolojisi

Çalışmamızda kullandığımız diğer bir teknoloji de Docker teknolojisidir. Şekil 6’da Docker üzerinde çalışan bir sistem mimarisi örnek olarak gösterilmektedir. Bu sistemde Docker üzerinde çalıştırılan ürün aradaki misafir işletim sistemine istek göndermekte, böylelikle uygulamalarda hız ve erişilebilirlik sağlanmaktadır (Özkök, 2020). Uygulamanın derleme ve test işlemleri Şekil 6’da sunulan örnek mimariye uygun olarak “docker” yazılım platformu üzerinde gerçekleştirilmiştir. Araştırma kapsamında bu yazılım platformunu kullanmak suretiyle uygulamada geliştirilen yazılım kodu “docker” yazılım platformu üzerinde test edilmiştir. Bu bağlamda araştırma kapsamında geliştirilen uygulamanın sunucu üzerinde kurulumu işlemleri de yine “docker” üzerinde konteynerleştirme aracı kullanılarak gerçekleştirilmiştir. Böylelikle araştırmada geliştirilen yazılıma ait kitaplıklar, sistem araçları, kod ve çalışma zamanı dahil olmak üzere çalışması için gerekli unsurları, "container" adlı standartlaştırılmış birimler halinde paketlenmiştir.



Şekil 6: Docker ve Sistem Mimarisi.

3.7 Bulut Bilişim (BB) Teknolojisi

Çalışmada bulut sunucuları kullanılmıştır. Bulut üzerinde çalışan sunucular gelişen teknoloji doğrultusunda internet sitelerini daha erişilebilir ve performanslı hale getirmek için seçilebilecek olanaklardan birisidir. Standart sunuculardan farklı olarak taşınabilirlik, ulaşılabilirlik gibi alanlarda eşsiz imkânlar sağlar. Donanım arızalarından etkilenmeden isteklere göre genişleyebilen sistemler oldukları için maliyet açısından da fayda sağlarlar. Bulut sunucular bu hizmetin alındığı şirketin sunucuları üzerinde donanım arızası gibi maliyetlerden etkilenmeden kullanıcılara verdiğimiz hizmeti sorunsuz yürütmemizi sağlarlar. Yeni oluşabilecek alan ihtiyaçlarını ve genişleme ihtiyaçlarını sorunsuz sağlamamızı sağlar. İnternete bağlanabilen tüm cihazlardan sunucuya erişip düzenlemeler ve ayarlamalar yapılabilir. Sistemi kapatmadan kaynakları arttırıp azaltabilirsiniz. Maliyet açısından ise fiziksel sunuculara göre daha az maliyetli olmaları açısından tercih edilmektedir.

4. DENEYSEL ÇALIŞMALAR

Tablo 1'de çalışma kapsamında gerçekleştirilen deneylere ilişkin bilgiler yer almaktadır. Söz konusu deney sonuçları değerlendirildiğinde 3 numaralı deneye ait verinin doğruluk oranlarının en yüksek oran olarak elde edildiği görülmektedir.

Tablo 1. Yapılan deneyler ve tahmin çıktıları.

Deney Numarası	Kimlik Bilgisi	Doğruluk oranı (%)
Deney 1	Öğrenci 1	97,81

Deney 2	Öğrenci 2	95,76
Deney 3	Öğrenci 3	98,82
Deney 4	Öğrenci 4	96,88
Deney 5	Öğrenci 5	93,76
Deney 6	Öğrenci 6	98,67
Deney 7	Öğrenci 7	97,63
Deney 8	Öğrenci 8	94,89
Deney 9	Öğrenci 9	98,02
Deney 10	Öğrenci 10	95,99

Tablo 1’deki deneysel sonuçlar i5-10300H 2.50GHz işlemci ve 8GB RAM özelliklerine sahip bilgisayar kullanılarak elde edilmiştir. Şekil 7’de yoklama alma sistemi görülmektedir. Sağ tarafta görülen kamerada amaçlanan sistem her öğrenci sınıfa girerken kapıyı gören bir kamera koyulması derse giren öğrencilerin kaydedilmesi yönünde amaçlanmıştır.

Şekil 7’de oluşturulan sistem kullanılmadan önceki kullanıcı alışkanlıkları incelenmiştir. Kullanıcının yaşadığı sorunlar ve harcadığı vakit gözlemlenmiştir.

Gerçek Zamanlı Yoklama Sistemi

Eğitimci E-Posta Adresi:

Eğitimci ID:

Ders Kodu:



Ömer

Şekil 7: Yoklama alma sistemi ve kamera görüntüsü.

5. SONUÇ VE DEĞERLENDİRME

Günümüzde güvenlik, analiz, kullanıcı doğrulama gibi birçok uygulama alanında kullanılan önemli bir teknolojidir. Bu çalışmada bulut üzerinde Öğretilebilir Makine (ÖM) teknolojisi kullanılarak bir web tabanlı Yüz Tanıma Sistemi (YTS-ÖM) geliştirilmiştir. Çalışma kapsamında geliştirilen bu sistem öğrencilerin yüzleri yoklama sistemine kaydederek öğrenci kaydını tutmaktadır. Önerilen sistem Web uygulamalarına bağlanmak için internet bağlantısı ve internet tarayıcısı açabilen ve kameralı bir cihaz kullanmaktadır. Çalışma neticesinde YTS-ÖM’nin başarı oranı %99.8 olarak ölçülmüştür. Araştırmamızda, ÖM teknolojisinin kullanıcıları kendi YZ sınıflandırma modellerini oluşturma konusunda güçlendirdiği anlaşılmaktadır. Araştırmada elde edilen sonuçlar, kullanıcıların kendi DL modellerini kodlama yapmadan geliştirmelerine imkân sağlayan ÖM teknolojisi kullanılarak geliştirilen YTS modelinin etkinliğini ortaya koymaktadır. Bu çalışmanın özellikle güvenlik endişesi nedeniyle birçok araştırmacıyı ve kullanıcıyı cezbeden YTS’lerin kullanıcılar tarafından geliştirilmesi ve kullanıma alınması açısından YZ çalışma alanına katkı sağlayacağı değerlendirilmektedir.

KAYNAKÇA

Aghdam O. A., 2018, Unconstrained Face Recognition Under Mismatched Conditions, İstanbul Teknik Üniversitesi, Yüksek Lisans Tezi.

- Akın F., 2019, Derin Öğrenme Tabanlı Yüz Tanıma Sisteminin Geliştirilmesi, Fırat Üniversitesi, Yüksek Lisans Tezi.
- Al Bazzaz, F. Y. (2017). Far Distance Unmanned Aerial Vehicles Control And Object Detection using Internet of Things Network and Embeded Systems (Master's Thesis, Fen Bilimleri Enstitüsü).
- Aslan E. S., 2016, Kapalı Ortamlar İçin Dağıtık Mimarili İnsan Tanıma Sistemi, İstanbul Teknik Üniversitesi, Yüksek Lisans Tezi.
- Ayata, F., & Çavuş, H. (2022). Yüz Tanıma Sistemlerinde Kullanılan ESA, YGH-DVM ve DSA Algoritmalarının Performans Testleri. Fırat Üniversitesi Fen Bilimleri Dergisi, 34(1), 39-48.
- Ayhan Ö., 2013, Yüz Öznitelik Çıkarımı için Geliştirilmiş Aktif Şekil Modeli , İstanbul Teknik Üniversitesi, Yüksek Lisans Tezi
- Bilgiç, A., Kurban, O. C., & Yildirim, T. (2017, May). Face Recognition Classifier based on Dimension Reduction in Deep Learning Properties. In 2017 25th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J. & Chen, A. (2020, April). Teachable machine: Approachable Web-based tool for exploring machine learning classification. In Extended abstracts of the 2020 CHI conference on human factors in computing systems (pp. 1-8).
- Çambay V. Y., 2019, Derin Öğrenme Kullanılarak FPGA'lar ile Gerçek Zamanlı Nesne Algılama ve Tanıma, Fırat Üniversitesi, Yüksek Lisans Tezi
- Eldem, A., Eldem, H., & Palali, A. (2017). Görüntü işleme teknikleriyle yüz algılama sistemi geliştirme. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 6(2), 44-48.
- Eleyan G., 2010, Görüntü Çerçevelerinde Yüz Algılama ve Veritabanı ile Eşleme Yapılması, Ankara Üniversitesi, Yüksek Lisans Tezi
- Fawzy A. M. I., 2019, Sanal Gösterge Panelinde Derin Öğrenme Tabanlı Gerçek-Zamanlı Nesne Algılama Uygulaması, Kocaeli Üniversitesi, Lisans Tezi
- Filiz S., 2012, Siber Güvenlikte Biyometrik Sistemler ve Yüz Tanıma, Gazi Üniversitesi, Yüksek Lisans Tezi.
- Gökberk B., 2006, Three Dimensional Face Recognition, Boğaziçi Üniversitesi, Yüksek Lisans Tezi
- Hazar Ş.,2008, Yapay Zeka Uygulamaları: Yüz Tanıma, Beykent Üniversitesi, Yüksek Lisans Tezi
- Holat R., 2014, Yüz Bulma ve Tanıma Sistemleri Kullanarak Kimlik Tespitinin Yapılması, Düzce Üniversitesi, Yüksek Lisans Tezi.
- Kacorri, H. (2017). Teachable machines for accessibility. ACM SIGACCESS Accessibility and Computing, (119), 10-18.
- Karaman İ., 2019, Bulut Bilişim Tabanlı Gerçek Zamanlı Yüz Tanıma Sistemi, Kırıkkale Üniversitesi, Yüksek Lisans Tezi.
- Manav C., 2010, Görüntü İşleme Yardımı İle Kimlik Tespiti, Gazi Üniversitesi, Yüksek Lisans Tezi
- Noori A. M., 2019, ORB Özelliklerine Dayalı Yüz Tanıma Sistemi, Selçuk Üniversitesi, Yüksek Lisans Tezi.
- Özgür Özkök, Virtual Machine ve Docker Container Sistemlerinin Tanımı, “<https://www.ozgurozkok.com/virtual-machine-ve-docker-container-sistemlerinin-tanitimi/>”, Son Erişim Tarihi:(19.05.2020).

- Rashid E., 2018, Raspberry PI ile Gerçek Zamanlı Yüz Tanıma ve Kontrol Sistemi, Selçuk Üniversitesi, Yüksek Lisans Tezi.
- Saeed S. B., 2016, Face Recognition System Based on Pca-Wavelet and Support Vector Machines, Hasan Kalyoncu Üniversitesi, Yüksek Lisans Tezi
- Saray Çetinkaya, T. & Sertbaş, A. (2022). Derin Öğrenme Algoritmalarının GPU ve CPU Donanım Mimarileri Üzerinde Uygulanması ve Performans Analizi: Deneysel Araştırma. Avrupa Bilim ve Teknoloji Dergisi, (33) , 10-19.
- Sertkaya, O, 2019, Yüz Tanıma için İmgelerin Kalite Ölçümü ve İyileştirilmesi, İstanbul Teknik Üniversitesi, Yüksek Lisans Tezi, 13.
- Sütçüler, E. (2006). Gerçek zamanlı video görüntülerinden yüz bulma ve tanıma sistemi.
- Teachable Machine Google, “<https://teachablemachine.withgoogle.com/faq#Tools-and-Resources>”, Son Erişim Tarihi (23.05.2020).
- Tırkaz Ç., 2008, Aktif Görünüm Modeli Kullanarak Yüz Tanıma, Yıldız Teknik Üniversitesi, Yüksek Lisans Tezi.
- Topkaya İ. S., 2008, Video Görüntülerinden Yüz Tanıma, Yıldız Teknik Üniversitesi, Yüksek Lisans Tezi.
- Yazıcı İ. M., 2008, Gerçek Zamanlı Kiosk Yüz Tanıma Sistemi, Gebze Yüksek Teknoloji Enstitüsü, Yüksek Lisans Tezi.
- Yıldırım M., 2011, Yüz Tanıma, İstanbul Kültür Üniversitesi, Yüksek Lisans Tezi.
- Zhu, K. (2019). An educational approach to machine learning with mobile applications (Doctoral dissertation, Massachusetts Institute of Technology).