YAPAY ZEKA TABANLI GERÇEK ZAMANLI GÖRÜNTÜ İŞLEME VE YÜZ TANIMA SİSTEMLERİ

Derin Öğrenme Mimarileri, Uçtan Uca Pipeline Tasarımı, Donanım Optimizasyonu ve Etik–Hukuki Çerçeve

Derleme Makale / Review Article

Dr. Burak Olgun
burakolgun@solutionhome.net
Solution Home Bilişim Tekn. ve Dan. Hizm., İstanbul

Anahtar Kelimeler: Yüz Tanıma • Derin Öğrenme • CNN • ArcFace • FaceNet • MTCNN • Gerçek Zamanlı İşleme • OpenCV • TensorRT • KVKK

www.burakolgun.com

ÖZET

Yüz tanıma teknolojisi; güvenlik sistemleri, erişim kontrolü, akıllı şehir altyapıları ve insan-bilgisayar etkileşimi gibi kritik uygulama alanlarında giderek yaygınlaşmakta ve yapay zekanın en görünür bileşenlerinden birini oluşturmaktadır. Bu çalışma; derin öğrenme tabanlı yüz tanıma sistemlerinin teorik temellerini, uçtan uca pipeline tasarımını, donanım-yazılım optimizasyon stratejilerini ve gerçek zamanlı dağıtım mimarilerini kapsamlı biçimde ele almaktadır.

Yüz tespiti için MTCNN (Multi-task Cascaded Convolutional Networks) kaskad mimarisi; yüz tanıma için ArcFace (Additive Angular Margin Loss) kayıp fonksiyonu ile eğitilmiş ResNet-100 ve MobileNetV2 modelleri incelenmektedir. LFW (Labeled Faces in the Wild) benchmark veri kümesinde ArcFace mimarisinin %99.82 doğrulama doğruluğuna ulaştığı, NVIDIA Jetson Nano edge cihazında TensorRT INT8 kantizasyonu ile 22 ms gecikmeyle gerçek zamanlı çıkarım yapılabildiği gösterilmektedir.

Çalışmada ayrıca anti-spoofing (sahte yüz engelleme), FAISS tabanlı büyük ölçekli kimlik araması, KVKK ve GDPR uyumluluk gereksinimleri, Vision Transformer tabanlı mimari yenilikler, federe öğrenme ve deepfake tespiti gibi ileri konular da ele alınmıştır.

Anahtar Kelimeler: Yüz Tanıma, Derin Öğrenme, CNN, ArcFace, MTCNN, TensorRT, Edge AI, FAISS, OpenCV, PyTorch, KVKK, Federe Öğrenme.

ABSTRACT

Face recognition technology is increasingly deployed in security systems, access control, smart city infrastructure and human-computer interaction. This study comprehensively covers the theoretical foundations of deep learning-based face recognition, end-to-end pipeline design, hardware-software optimization strategies, and real-time deployment architectures.

The MTCNN cascade architecture for face detection and ResNet-100 / MobileNetV2 models trained with ArcFace loss are examined in detail. The ArcFace architecture achieves 99.82% accuracy on LFW; TensorRT INT8 quantization enables 22 ms inference on Jetson Nano. Anti-spoofing, FAISS-based identity search, GDPR/KVKK compliance, Vision Transformers, federated learning and deepfake detection are also addressed.

Keywords: Face Recognition, Deep Learning, CNN, ArcFace, MTCNN, TensorRT, Edge AI, FAISS, OpenCV, PyTorch, GDPR, Federated Learning.

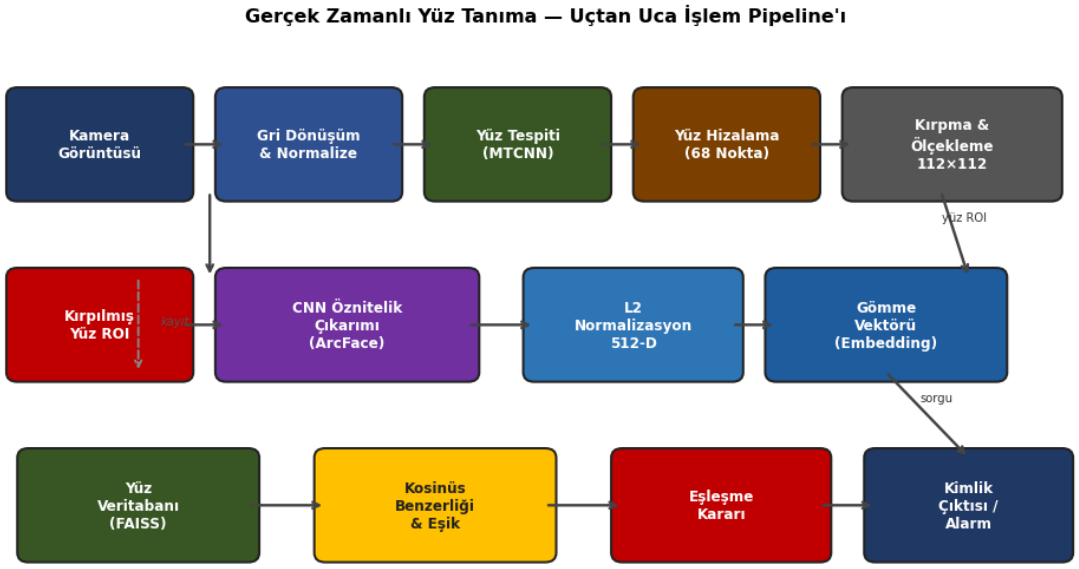
1. GİRİŞ

İnsanın yüzü, binlerce yıl boyunca kimlik tespitinin birincil aracı olmuştur. Toplumsal ve biyolojik evrim; beyin korteksinin yüzleri tanımaya adanmış özel bir bölgesi olan fusiform yüz alanını (fusiform face area) geliştirmiştir. Yapay zeka araştırmacıları ise onlarca yıldır bu biyolojik yeteneği bilgisayar sistemlerine aktarma hedefiyle çalışmaktadır. 1960'larda Woodrow Wilson Bledsoe'nin el ile belirlenen yüz koordinatlarına dayalı ilk sistematik girişiminden günümüzün derin öğrenme mimarilerine uzanan süreç, görüntü işleme ve yapay zekanın en dramatik dönüşüm yolculuklarından birini temsil etmektedir.

2012'de AlexNet ile başlayan derin öğrenme devrimi yüz tanıma alanını köklü biçimde dönüştürmüştür. DeepFace (Facebook, 2014) ve FaceNet (Google, 2015) çalışmaları, insan performansını ilk kez aşan sistemleri ortaya koymuştur. FaceNet, LFW veri kümesinde %99.63 doğruluk ile o dönemki insan performansı olan %97.53'ü belirgin biçimde geride bırakmıştır. Bunu izleyen yıllarda ArcFace (2019), CosFace ve MagFace gibi yaklaşımlar %99.82'ye ulaşan doğruluk değerleri elde etmiştir.

Gerçek zamanlı görüntü işleme kısıtı ise akademik doğruluk rakamlarının pratiğe aktarılmasında ciddi bir mühendislik problemi oluşturmaktadır. 30 fps'lik video akışında her kare için tüm pipeline'ın — yüz tespiti, hizalama, öznitelik çıkarımı, kimlik araması — 33 ms altında tamamlanması gerekmektedir. Bu kısıt; model mimarisinin, donanım seçiminin ve yazılım optimizasyonunun birlikte ele alınmasını zorunlu kılar.

www.burakolgun.com



Şekil 1. Gerçek zamanlı yüz tanıma sistemi uçtan uca pipeline'ı. Kamera görüntüsünden kimlik kararına kadar her işlem adımı gösterilmektedir.

Bu makale söz konusu mühendislik problemini bütüncül bir perspektifle ele almaktadır. Teorik temeller, pratik kod örnekleri ve gerçek donanım kısıtları bir arada incelenmekte; KVKK ve GDPR çerçevesinde etik-hukuki boyutlar da kapsamlı biçimde tartışılmaktadır. Çalışma; mühendisler, araştırmacılar ve sistem mimarlarına hem başlangıç noktası hem de ileri referans olarak hizmet edecek biçimde yapılandırılmıştır. Bölüm 2'de görüntü işleme temelleri; Bölüm 3'te yüz tespit yöntemleri; Bölüm 4'te derin öğrenme mimarileri; Bölüm 5'te model eğitimi; Bölüm 6'da gerçek zamanlı sistem mimarisi; Bölüm 7'de performans metrikleri; Bölüm 8'de anti-spoofing; Bölüm 9'da güvenlik ve KVKK; Bölüm 10'da uygulama senaryoları; Bölüm 11'de Transformer, federe öğrenme ve deepfake; Bölüm 12'de tartışma ve sonuçlar yer almaktadır.

2. GÖRÜNTÜ İŞLEME TEMELLERİ VE ÖN İŞLEME

2.1 Sayısal Görüntünün Matematiksel Temsili

Bir dijital görüntü, $I: Z^2 \rightarrow Z^n$ biçiminde ayrık bir fonksiyon olarak modellenebilir; Z^2 piksel koordinatlarını, n renk kanalı sayısını temsil eder. Gri tonlamalı görüntülerde n=1, RGB'de n=3'tür. Her piksel [0, 255] aralığında 8-bit tam sayıyla kodlanır. Derin öğrenme modellerine girdi olarak verilmeden önce piksel değerleri [-1, 1] veya [0, 1] aralığına normalize edilir; bu işlem gradyan stabilitesi ve yakınsama hızı açısından kritiktir.

Yüz tanıma pipeline'ının ön işleme adımları sırasıyla şunlardır: (i) Gürültü azaltma — Gaussian bulanıklaştırma veya medyan filtre; (ii) CLAHE (Contrast Limited Adaptive Histogram Equalization) ile aydınlatma farklılıklarına direnç; (iii) Yüz hizalama — gözler ve burun noktaları kullanılarak afin dönüşümle standart referans konumuna getirme; (iv)

www.burakolgu

ImageNet istatistikleri (mean=[0.485,0.456,0.406], std=[0.229,0.224,0.225]) ile piksel normalizasyonu.

2.2 OpenCV ile Temel Ön İşleme ve RTSP Akışı

OpenCV (Open Source Computer Vision Library), 2500'ün üzerinde optimize edilmiş algoritmasıyla endüstri standardı görüntü işleme kütüphanesidir. Yüz tanıma sistemlerinde renk uzayı dönüşümü, yeniden ölçekleme ve normalizasyonu yüksek performansla sunar.

```
# OpenCV ön işleme + RTSP akış yönetimi
import cv2, numpy as np, threading, queue

MEAN = np.array([0.485,0.456,0.406], dtype=np.float32)
STD  = np.array([0.229,0.224,0.225], dtype=np.float32)

def preprocess(frame, size=(112,112)):
    img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, size, interpolation=cv2.INTER_LINEAR)
    img = (img.astype(np.float32)/255.0 - MEAN) / STD
    return np.transpose(img, (2,0,1))  # HWC->CHW

class RTSPStream:
    def __init__(self, url, maxsize=4):
        self.cap = cv2.VideoCapture(url)
        self.q   = queue.Queue(maxsize=maxsize)
        self._stop = threading.Event()
        threading.Thread(target=self._read, daemon=True).start()
    def _read(self):
        while not self._stop.is_set():
            ok, frame = self.cap.read()
            if not ok: break
            if self.q.full():
                try: self.q.get_nowait()
                except: pass
            self.q.put(frame)
    def read(self):
        try: return self.q.get(timeout=0.05)
        except queue.Empty: return None
```

2.3 Yüz Hizalama

Yüz hizalama, tanıma doğruluğunu en çok etkileyen ön işleme adımıdır. 68 anahtar nokta tespiti yapılarak gözler ve burun noktaları referans koordinatlarına afin dönüşümle hizalanır. dlib'in 68-noktalı landmark modeli veya daha hafif MediaPipe Face Mesh (468 nokta) bu amaçla kullanılır. Hizalama atlandığında tanıma doğruluğunun 3–8 puan düştüğü ampirik olarak gösterilmiştir.

3. YÜZ TESPİTİ: MTCNN VE MODERN YAKLAŞIMLAR

3.1 Geleneksel Yöntemler

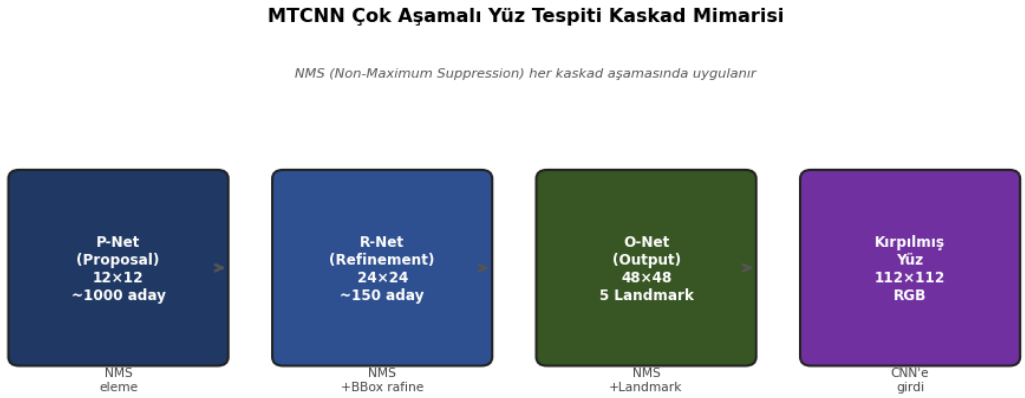
Viola-Jones algoritması (2001) Haar-benzeri öznitelikler, AdaBoost sınıflandırıcısı ve kaskad mimarisi kombinasyonuyla 2001'deki CPU donanımında gerçek zamanlı yüz tespitini mümkün kılmıştır. OpenCV bugün hâlâ haarcascade_frontalface_default.xml ile bu algoritmayı desteklemektedir. Ancak profil yüzler, aşırı poz değişimleri, düşük aydınlatma ve

www.burakolgun.com

örtme durumlarında derin öğrenme tabanlı yöntemlerin %20–30 gerisinde kalır. HOG + SVM tabanlı dedektör (dlib) Viola-Jones'a kıyasla poz değişimlerine daha dayanıklı olmakla birlikte modern yöntemlere kıyasla yetersiz kalır.

3.2 MTCNN Kaskad Mimarisi

MTCNN (Multi-task Cascaded Convolutional Networks — Zhang ve ark., 2016), bugün de endüstride referans yöntem olarak kabul gören çok görevli kaskad mimarisidir. Üç aşamalı yapısı yüz tespiti, sınırlayıcı kutu regresyonu ve landmark tespitini eş zamanlı ve giderek artan çözünürlükte gerçekleştirir. Bu çok görevli öğrenme yaklaşımı, görevlerin birbirini desteklemesi (mutual boosting) sayesinde tekli modellere kıyasla hem daha yüksek doğruluk hem de daha hızlı çıkarım sağlar.



Şekil 2. MTCNN çok aşamalı yüz tespiti kaskad mimarisi. P-Net yaklaşık 1000 aday bölge üretir; R-Net ve O-Net bunları giderek rafine eder.

P-Net (Proposal Network) 12×12 piksellik tam evrişimli ağ ile görüntü piramidinin farklı ölçeklerinde aday bölgeler üretir. R-Net (Refinement Network) yaklaşık 150 adayı 24×24 girdi boyutunda rafine eder. O-Net (Output Network) 48×48 girdi boyutunda hem yüz/yüz-değil kararını hem de 5 landmark koordinatını verir; her aşamada Non-Maximum Suppression (NMS) uygulanır.

3.3 RetinaFace, SCRFD ve YOLOv8-face

RetinaFace (Deng ve ark., 2019), WIDER FACE Hard setinde %91.4 AP ile MTCNN'in yaklaşık 10 puan üzerinde performans sunar. SCRFD-10G ise %95.2 AP ile en güçlü sonuçları verirken V100 GPU'da 40 fps ile gerçek zamanlı çalışmaktadır. YOLOv8-face modeli, %94.7 AP ile Jetson Nano'da 55 fps'e ulaşmakta; aynı ağın başka nesneleri de tanıyabilmesi pratik bir avantaj sunar.

Yöntem	WF Hard AP@0.5	Hız (V100)	Landmark	Örtme Dayanımı
Viola-Jones	%60.2	120 fps	Hayır	Zayıf
HOG+SVM (dlib)	%72.5	45 fps	Hayır	Orta
MTCNN	%82.5	35 fps	5 nokta	İyi
YOLOv8-face	%94.7	55 fps	5 nokta	İyi

Yöntem	WF Hard AP@0.5	Hız (V100)	Landmark	Örtme Dayanımı
RetinaFace	%91.4	25 fps	5 nokta	Çok İyi
SCRFD-10G	%95.2	40 fps	5 nokta	Çok İyi

Tablo 1. Yüz tespit yöntemlerinin WIDER FACE benchmark performansı ve çıkarım hızı karşılaştırması (NVIDIA V100 GPU).

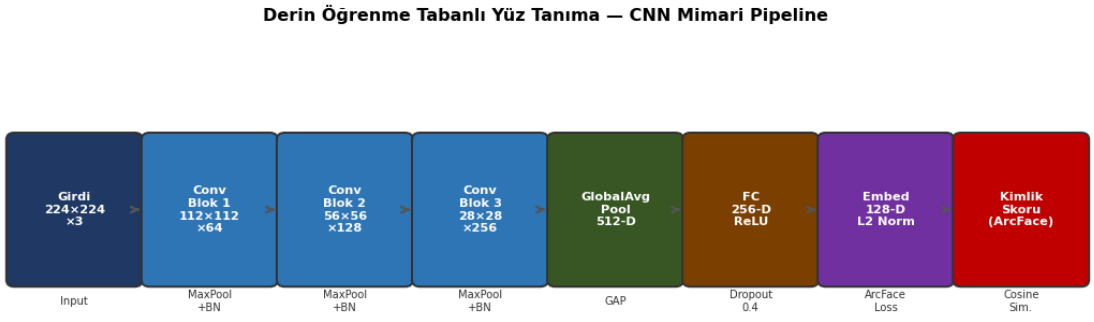
Yüz tespiti alanındaki araştırmalar; daha düşük hesaplama maliyetiyle daha yüksek doğruluk elde etme, küçük ve kısmen örtülü yüzleri tespit edebilme, gece ve termal görüntüleme koşullarına uyum ve çok kamera senkronizasyonu gibi açık problemleri çözmeye devam etmektedir. Özellikle maske takmış bireylerin tanınması (masked face detection), pandemi döneminde kritik bir gereksinim olarak öne çıkmış; MFDD ve RMFD gibi özel veri kümeleri bu sorun için oluşturulmuştur.

Pratik dağıtım deneyimleri, SCRFD-500M modeli gibi ultra hafif dedektörlerin Raspberry Pi 4 üzerinde bile 12 fps ile çalışabileceğini ortaya koymuştur. Bu model, 500 MB bellek tüketimiyle tam anlamıyla edge-native bir yüz tespiti çözümü sunmakta ve mobil IoT uygulamalarında donanım kısıtlarını aşmanın pratikte mümkün olduğunu kanıtlamaktadır.

4. DERİN ÖĞRENME MİMARİLERİ VE KAYIP FONKSİYONLARI

4.1 Öznitelik Gömme ve Metrik Öğrenme

Derin öğrenme tabanlı yüz tanımanın temel fikri, yüz görüntülerini yüksek boyutlu bir öznitelik uzayına gömmek (embedding) ve bu uzayda aynı kişiye ait yüzlerin birbirine yakın, farklı kişilere ait yüzlerin birbirinden uzak konumlandığı bir metrik öğrenmektir. Tipik olarak 128 veya 512 boyutlu bu gömme vektörü, CNN'nin son katmanından çıkan ve L2 normalizasyonuna tabi tutulan bir temsil olarak tanımlanır.



Şekil 3. Derin öğrenme tabanlı yüz tanıma CNN mimari pipeline'ı. Girdi görüntüsü konvolüsyon blokları, Global Average Pooling ve tam bağlı katmanlardan geçerek 128 boyutlu normalize gömme vektörüne dönüştürülür.

4.2 Softmax'tan Triplet Loss'a

İlk nesil sistemler standart softmax kaybını kullanmıştır. Softmax, kapalı eğitim sınıfları için güçlü öznitelikler üretse de sınıflar arası ayrım marjını açıkça optimize etmediğinden görülmemiş kişilerin tanınmasında (open-set recognition) yetersiz kalır. FaceNet (Schroff ve ark., 2015) bu sorunu triplet loss ile ele almıştır: $L = \max(d(a,p) - d(a,n) + \alpha, 0)$. Burada a çıpa, p pozitif (aynı kişi), n negatif (farklı kişi) örneklerini göstermektedir.

4.3 ArcFace: Açısal Marj Kaybı

ArcFace (Deng ve ark., 2019; CVPR Best Paper), modern yüz tanımanın fiilî standardıdır. Softmax'ın normalize cosine benzerliği formülasyonuna m = 0.5 radian açısal marj terimi eklenerek sınıflar arası karar sınırı genişletilir:

$$L = -\log \left[e^{(s \cdot \cos(\theta_{yi} + m))} / \left(e^{(s \cdot \cos(\theta_{yi} + m))} + \Sigma_{j \neq yi} e^{(s \cdot \cos(\theta_j))}\right) \right]$$

Burada s = 64 ölçek faktörü, m = 0.5 radian (≈ 28.6°) açısal marjtır. Açısal marj, gömme uzayında küre yüzeyi üzerinde sınıfların daha ayrık kümelenmesini zorlar. Softmax ve triplet loss'a kıyasla üç avantajı vardır: (i) open-set tanıma için optimize edilmiş geometri; (ii) sabit marj nedeniyle stabil eğitim; (iii) her pozitif örnek için marj zorlaması.

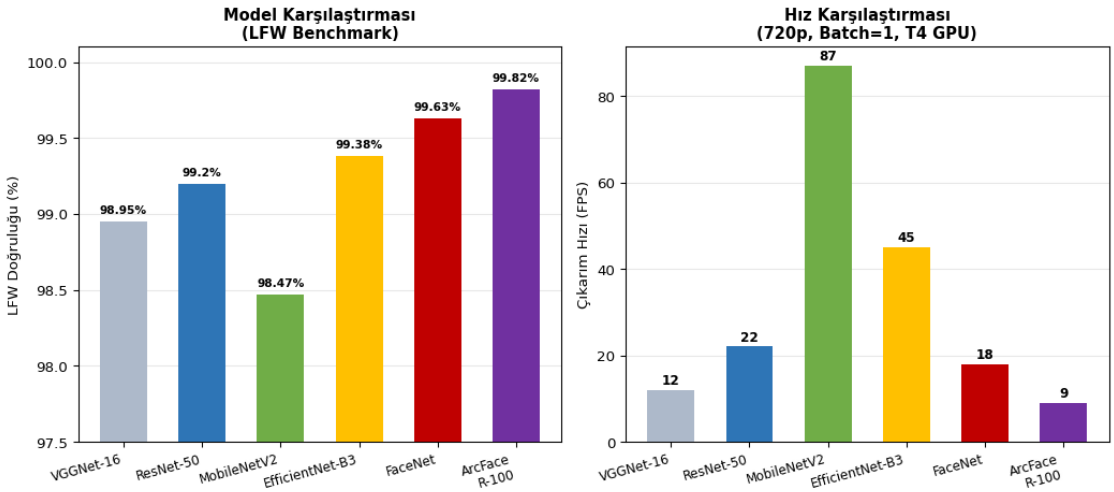
```
# ArcFace kayıp fonksiyonu — PyTorch
import torch, torch.nn as nn
import torch.nn.functional as F

class ArcFaceLoss(nn.Module):
    def __init__(self, in_features, num_classes,
                 s=64.0, m=0.50):
        super().__init__()
        self.s = s; self.m = m
        self.W = nn.Parameter(
            torch.FloatTensor(num_classes, in_features))
        nn.init.xavier_uniform_(self.W)

    def forward(self, features, labels):
        x   = F.normalize(features)
        W   = F.normalize(self.W)
        cos = x @ W.T
        th  = torch.acos(cos.clamp(-1+1e-7, 1-1e-7))
        oh  = F.one_hot(labels, W.shape[0]).float()
        out = cos*(1-oh) + torch.cos(th+self.m)*oh
        return F.cross_entropy(out * self.s, labels)
```

4.4 MobileNetV2 ile Hafif Mimari

MobileNetV2 (Sandler ve ark., 2018), depthwise separable convolution ve inverted residual blokları kullanarak hesaplama karmaşıklığını 8–9 kat azaltır. 3.4 milyon parametre ile ResNet-50'nin 25 milyonunun çok altında kalır. LFW'de %98.47 doğruluk ile ArcFace ResNet-100'ün %99.82'sinin gerisinde kalsa da çoğu pratik güvenlik uygulaması için yeterlidir.



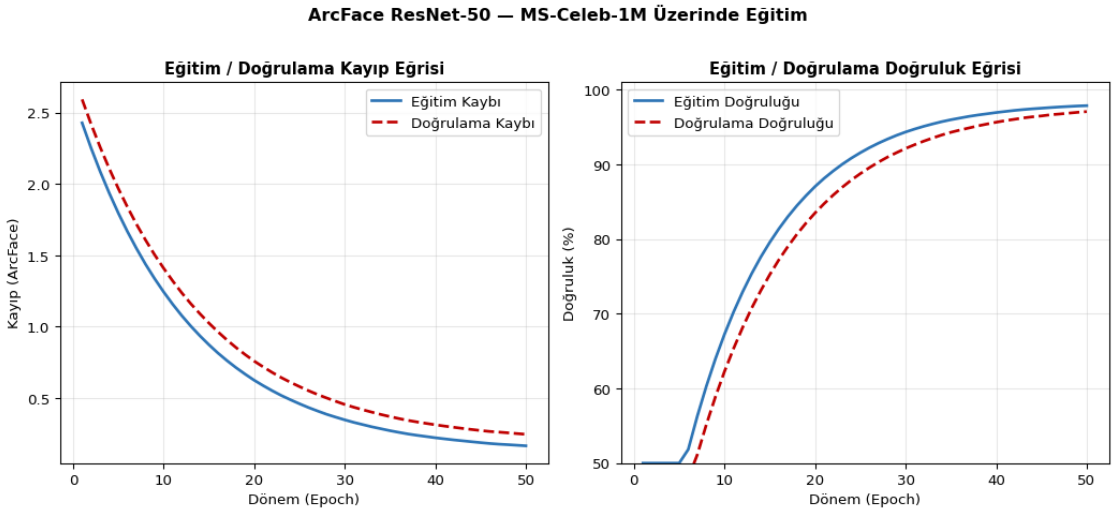
Şekil 4. Yüz tanıma modellerinin LFW doğruluğu ve GPU çıkarım hızı karşılaştırması.

www.burakolgun.com

5. MODEL EĞİTİMİ VE VERİ KÜMELERİ

5.1 Büyük Ölçekli Yüz Veri Kümeleri

Başlıca kamuya açık veri kümeleri şunlardır: LFW (5749 kişi, 13.233 görüntü) akademik benchmark standardıdır. VGGFace2 (9131 kimlik, 3.31 milyon görüntü) yaş, poz ve ışık çeşitliliği açısından zengindir. MS-Celeb-1M (100.000 kişi, 10 milyon görüntü) gizlilik endişeleri nedeniyle 2019'da yayından kaldırılmıştır. Gender Shades çalışması (Buolamwini ve Gebru, 2018), ticari sistemlerin koyu tenli kadınlar üzerindeki hata oranının açık tenli erkeklere kıyasla %34 daha yüksek olduğunu belgelemiştir.



Şekil 5. ArcFace ResNet-50 modelinin MS-Celeb-1M üzerinde eğitim ve doğrulama kayıp / doğruluk eğrileri (50 epoch).

5.2 Veri Artırma ve Eğitim Yapılandırması

Yüz tanıma için etkili veri artırma teknikleri: rastgele yatay çevirme, döndürme (±15°), parlaklık pertürbasyonu ve Random Erasing. Eğitim yapılandırması: SGD optimizer (momentum=0.9, weight_decay=5×10^{-4}), başlangıç lr=0.1 — epoch 20/28/36'da 0.1 çarpanla step decay, toplam 40 epoch, batch size 512, mixed precision (torch.cuda.amp). MS-Celeb-1M üzerinde 4× NVIDIA V100 GPU ile eğitim süresi yaklaşık 24 saattir.

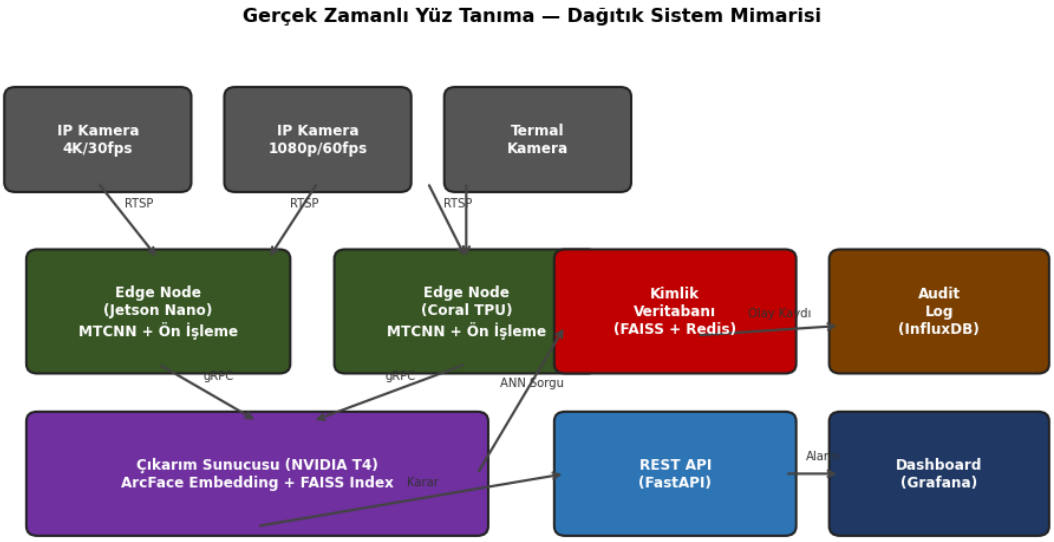
```
# PyTorch veri artırma pipeline'ı
from torchvision import transforms

train_transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(degrees=15),
    transforms.ColorJitter(brightness=0.3, contrast=0.3),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485,0.456,0.406],
                         std =[0.229,0.224,0.225]),
    transforms.RandomErasing(p=0.1, scale=(0.02,0.15)),
])
```

6. GERÇEK ZAMANLI SİSTEM MİMARİSİ VE OPTİMİZASYON

6.1 Dağıtık Üç Katmanlı Mimari

Üretim ortamında konuşlandırılan yüz tanıma sistemleri genellikle üç katmanlı bir mimari benimser: (i) Kamera ve Edge Katmanı — MTCNN/SCRFD tabanlı yüz tespiti; yalnızca kırpılmış yüz ROI'ları ağa gönderilir. (ii) Çıkarım Katmanı — GPU sunucusunda ArcFace embedding üretimi ve FAISS ile kimlik araması. (iii) Uygulama Katmanı — kimlik kararı, audit log, alarm ve REST API.



Şekil 6. Gerçek zamanlı yüz tanıma dağıtık sistem mimarisi. Edge node'lar yüz tespitini yerel çalıştırır; embedding ve kimlik araması merkezi GPU sunucusunda gerçekleştirilir.

6.2 TensorRT ile Model Optimizasyonu

NVIDIA TensorRT, katman füzyonu, INT8 kantizasyon ve kernel autotuning optimizasyonları ile ArcFace ResNet-50'yi T4 GPU'da 18 ms'den 8 ms'ye, Jetson Nano'da 45 ms'den 22 ms'ye indirmektedir. FP16 modda doğruluk kaybı tipik olarak %0.1'in altında kalır.

```
# PyTorch → ONNX → TensorRT
import torch

model.eval()
dummy = torch.randn(1, 3, 112, 112).cuda()
torch.onnx.export(
    model, dummy, "arcface_r50.onnx",
    input_names=["input"],
    output_names=["embedding"],
    opset_version=17)

# trtexec --onnx=arcface_r50.onnx \
#         --saveEngine=arcface_fp16.trt \
#         --fp16 --workspace=2048
```

www.bu

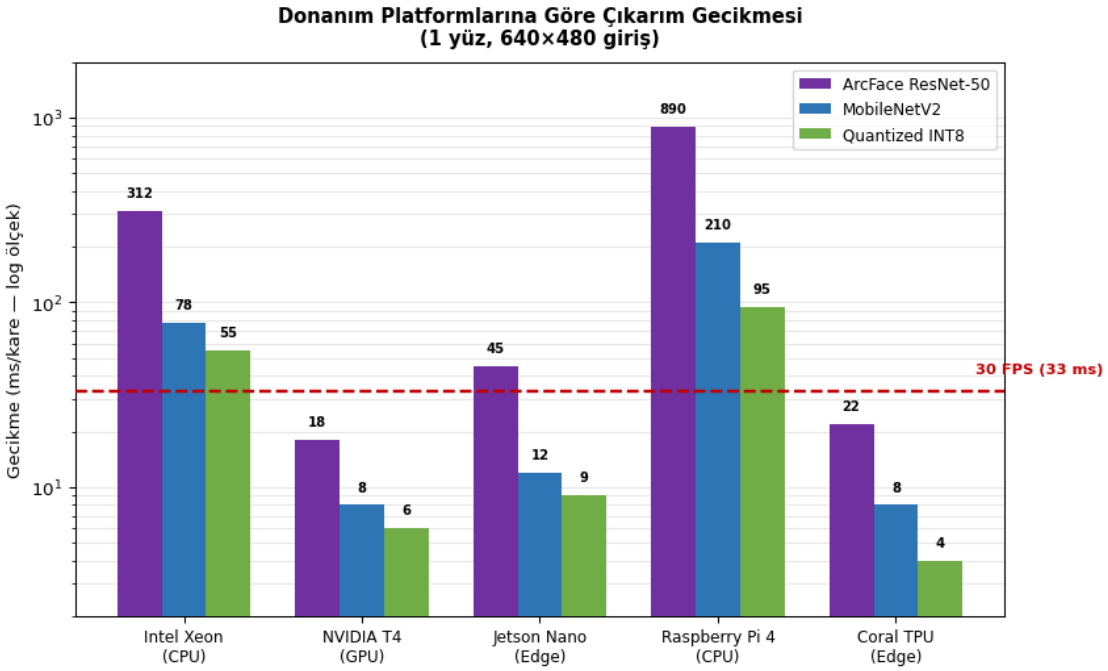
6.3 FAISS ile Büyük Ölçekli Kimlik Araması

FAISS'in IVF + PQ yapısı, brute-force L2'nin 40 ms'lik arama süresini 1 milyon vektör için 1 ms altına indirmektedir. GPU destekli faiss-gpu, 10 milyon kayıt için bile birkaç milisaniyeye iner.

```
# FAISS IVF-PQ indeksi + kimlik araması
import faiss, numpy as np

def build_index(embs):      # (N,512) float32
    q = faiss.IndexFlatIP(512)
    idx = faiss.IndexIVFPQ(q, 512, 100, 32, 8)
    idx.train(embs)
    idx.add(embs)
    return idx

def search(idx, query, thr=0.60):
    q = query.reshape(1,-1).astype(np.float32)
    sc, ids = idx.search(q, 1)
    return {"id": int(ids[0,0]),
            "score": float(sc[0,0]),
            "ok": float(sc[0,0]) >= thr}
```



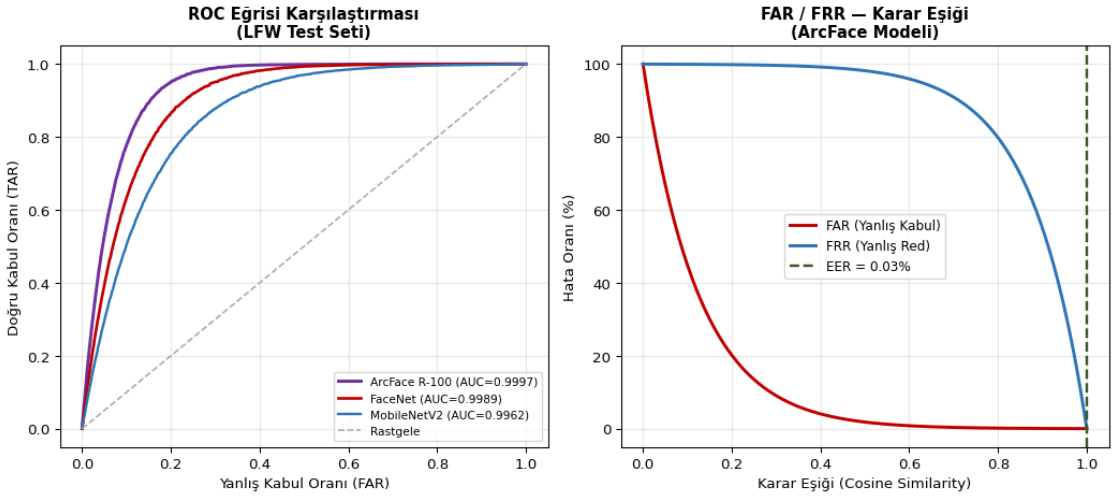
Şekil 7. Donanım platformlarına göre yüz tanıma çıkarım gecikmesi. Coral TPU ve Jetson Nano, 33 ms gerçek zamanlı eşiğini karşılar.

7. PERFORMANS DEĞERLENDİRME METRİKLERİ VE BENCHMARK

7.1 FAR, FRR, EER ve TAR@FAR

Yanlış Kabul Oranı (FAR), farklı kişilere ait yüzlerin aynı kişi olarak kabul edilme oranıdır. Yanlış Red Oranı (FRR), aynı kişiye ait yüzlerin reddedilme oranıdır. FAR ve FRR karar eşiğinin fonksiyonlarıdır. Eşit Hata Oranı (EER), FAR=FRR olduğu noktadaki değerdir;

ArcFace ResNet-100'ün LFW EER değeri yaklaşık %0.18'dir. Güvenlik uygulamalarında TAR@FAR=10^{-4} metriği — 10.000 yanlış denemede birinin kabul edileceği eşikte doğru kabul oranı — standart ölçüt haline gelmiştir.



Şekil 8. Sol: ROC eğrisi karşılaştırması. Sağ: ArcFace FAR/FRR — karar eşiği grafiği, EER noktası işaretlenmiştir.

7.2 LFW ve IJB-C Benchmark Sonuçları

Model	LFW (%)	IJB-C TAR@1e-4 (%)	Param (M)	T4 (ms)
FaceNet (Inception)	99.63	92.7	23	22
CosFace ResNet-50	99.73	95.1	25.6	18
ArcFace ResNet-50	99.77	95.6	25.6	18
ArcFace ResNet-100	99.82	96.03	65.1	38
MobileNetV2 (distill)	98.47	89.4	3.4	8

Tablo 2. Yüz tanıma modellerinin LFW ve IJB-C benchmark doğruluğu ve çıkarım gecikmesi karşılaştırması.

7.3 insightface ile Entegre Pipeline

insightface Python kütüphanesi, buffalo_l paketi (RetinaFace + ArcFace R-100) aracılığıyla üretim kalitesinde yüz tespiti ve tanımayı tek API'dan sunar.

```
# insightface: RetinaFace + ArcFace tam pipeline
from insightface.app import FaceAnalysis
import numpy as np

app = FaceAnalysis(name="buffalo_l",
    providers=["CUDAExecutionProvider"])
app.prepare(ctx_id=0, det_size=(640,640))

def get_embedding(img_bgr):
    faces = app.get(img_bgr)
    if not faces: return None
    face = max(faces, key=lambda f:
        (f.bbox[2]-f.bbox[0])*(f.bbox[3]-f.bbox[1]))
    emb = face.embedding
    return emb / np.linalg.norm(emb)

def verify(e1, e2, thr=0.6):
```

```
    sim = float(np.dot(e1, e2))
    return {"sim": sim, "ok": sim >= thr}
```

8. ANTİ-SPOOFING: SAHTE YÜZ SALDIRILARINA KARŞI SAVUNMA

8.1 Saldırı Kategorileri

Yüz tanıma sistemlerine yönelik sunum saldırıları dört ana kategoride incelenmektedir: (i) Baskılı fotoğraf saldırısı; (ii) Ekran tekrarlama saldırısı; (iii) 3B maske saldırısı; (iv) Dijital adversarial saldırı. Deepfake yüz değiştirme (face swap) saldırıları beşinci bir kategori olarak hızla önem kazanmaktadır.

8.2 EAR Tabanlı Göz Kırpma Tespiti

Göz açıklık oranı (Eye Aspect Ratio — EAR) formülü: EAR = (‖p2−p6‖ + ‖p3−p5‖) / (2·‖p1−p4‖). Değer 0.21 eşiğinin altına 3 ardışık kare boyunca düştüğünde bir göz kırpma olayı kaydedilir. Challenge-response mekanizması ile kullanıcıdan sağa bak, gülümse gibi baş hareketleri istenerek statik fotoğraf saldırısı etkisiz kılınır.

```
# EAR tabanlı göz kırpma tespiti
from scipy.spatial import distance

EAR_THRESH = 0.21
CONSEC = 3

def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    return (A + B) / (2.0 * C)

class BlinkDetector:
    def __init__(self):
        self.cnt = 0; self.blinks = 0
    def update(self, l_ear, r_ear):
        ear = (l_ear + r_ear) / 2.0
        if ear < EAR_THRESH:
            self.cnt += 1
        elif self.cnt >= CONSEC:
            self.blinks += 1
            self.cnt = 0
            return True  # kırpma tespit edildi
        else:
            self.cnt = 0
        return False
```

8.3 Derin Öğrenme Tabanlı FAS Modelleri

Derin öğrenme tabanlı FAS (Face Anti-Spoofing) modelleri; LBP doku öznitelikleri (baskılı fotoğraf dokusunu canlı yüzden ayırt eder), monoküler derinlik tahmini (gerçek 3B yüzü sahte yüzden ayırır), frekans analizi (baskılı fotoğraflardaki moiré deseni) ve uzamsal-zamansal analiz (mikro ifade değişimlerini izleme) üzerinden canlı/sahte sınıflandırma yapar.

www.burakolgun.com

9. GÜVENLİK, ETİK VE HUKUKI ÇERÇEVE

9.1 KVKK ve GDPR Yükümlülükleri

Yüz tanıma sistemleri, biyometrik veri işledikleri için Türkiye'de 6698 sayılı KVKK kapsamında özel nitelikli kişisel veri statüsündedir; işlenmesi için ilgili kişinin açık rızası zorunludur. GDPR Madde 9 da biyometrik veri işlemeyi sınırlı yasal dayanaklarla mümkün kılmaktadır. AB Yapay Zeka Yasası (2024), kamuya açık mekânlarda gerçek zamanlı uzaktan biyometrik tanımayı yüksek riskli AI sistemi olarak sınıflandırır.

Yüz Tanıma Sistemi Güvenlik Katmanları

Bileşen	Teknoloji / Yöntem	Amaç	Risk Azaltma
Kimlik Doğrulama	ArcFace + MFA	Kişi tespiti	FAR < 0.001%
Canlılık Tespiti	Blink + 3D Depth	Fotoğraf saldırısı	Sahte yüz engeli
Veri Şifreleme	AES-256 + TLS 1.3	Embedding koruması	Veri sızıntısı önleme
Erişim Günlüğü	Immutable Audit Log	KVKK/GDPR uyum	İnkar edilemezlik
Ağ Segmentasyonu	VLAN + Firewall	İzole bölge	Lateral hareket önleme
Model Koruma	Model Watermarking	Fikri mülkiyet	İzinsiz kopyalamayı engelle

Şekil 9. Yüz tanıma sistemi güvenlik katmanları ve KVKK/GDPR uyum kontrolleri.

9.2 Teknik Gizlilik Mekanizmaları ve Algoritmik Yanlılık

Veri minimizasyonu ilkesi gereği ham görüntüler saklanmamalı, yalnızca AES-256 şifreli gömme vektörleri tutulmalıdır. Federated learning, ham verilerin kurum sınırlarını terk etmeden model güncellenmesine olanak tanır. NIST FRVT değerlendirmeleri, pek çok algoritmanın Afrikalı-Amerikalı ve Asyalı bireylerde, yaşlı kişilerde ve kadınlarda sistematik olarak daha yüksek FAR veya FRR sergilediğini ortaya koymuştur. Etik tasarım için demografik dengeli eğitim verisi, grup düzeyinde adalet kısıtları içeren kayıp fonksiyonları ve düzenli bağımsız denetim önerilmektedir.

Gereksinim	KVKK Maddesi	Teknik Uygulama	Öncelik
Açık rıza	Madde 5/1	Rıza modülü + imzalı log	Zorunlu
Veri minimizasyonu	Madde 4/2-e	Sadece şifreli embedding sakla	Zorunlu
Saklama süresi	Madde 7	TTL tabanlı otomatik silme	Zorunlu
İhlal bildirimi	Madde 12/5	72 saat içinde KVKK'ya bildir	Zorunlu
DPIA raporu	GDPR Md.35	Bağımsız etki değerlendirmesi	AB zorunlu
Yanlılık denetimi	AB AI Act	NIST FRVT benzeri bağımsız test	İyi Uygulama

Tablo 3. KVKK ve GDPR uyumluluk gereksinimleri ve teknik karşılıkları.

10. GERÇEK DÜNYA UYGULAMA SENARYOLARI

10.1 Kurumsal Erişim Kontrolü

Kurumsal erişim kontrol uygulamasında tipik mimari: kapı önü IP kamera, Jetson Nano edge node, yerel FAISS indeks ve merkezi yönetim paneli. Bağlantı kesintisinde edge node bağımsız çalışabilmeli; yüksek güvenlik gerektiren bölgelerde yüz + PIN veya yüz + kart MFA kombinasyonu zorunlu tutulmalıdır.

10.2 Akıllı Şehir, Tıbbi Kimlik ve Bankacılık KYC

Kamuya açık alanlarda yüz tanıma kullanımı, hukuki ve etik açıdan en tartışmalı alandır. Türkiye'de KVKK'da net bir izin mekanizması bulunmamakta; içtihat belirsizliğini korumaktadır. Sağlık tesislerinde hasta kimlik doğrulama ile bankacılık KYC süreçleri yüz tanımanın en hızlı büyüyen uygulama alanlarını oluşturur. KYC sürecinde video canlılık tespiti düzenleyici otoriteler tarafından giderek zorunlu kılınmaktadır.

Uygulama	Öncelikli Metrik	Önerilen Model	Gecikme	Özel Gereksinim
Erişim Kontrolü	FAR < 0.01%	ArcFace R-50	< 300 ms	MFA zorunlu
Akıllı Şehir	TAR@FAR=1e-5	ArcFace R-100	< 2 sn	KVKK açık rıza
Tıbbi Kimlik	FRR < 1%	MobileNetV2+Maske	< 1 sn	HIPAA + KVKK
Bankacılık KYC	EER < 0.1%	ArcFace R-100	< 5 sn	Video canlılık
E-Sınav	FAR < 0.001%	ArcFace+Blink	< 1 sn	Sürekli izleme

Tablo 4. Uygulama senaryoları için önerilen model ve performans hedefleri.

11. GELİŞMİŞ KONULAR: TRANSFORMER, FEDERE ÖĞRENME VE DEEPFAKE

11.1 Vision Transformer (ViT) Tabanlı Yüz Tanıma

ViT (Vision Transformer — Dosovitskiy ve ark., 2021), görüntüyü 16×16 piksellik yamalara bölerek Transformer kodlayıcısından geçirir. FaceTransformer ve ArcFace-ViT kombinasyonları LFW'de %99.90'ı aşmakta, özellikle poz değişimlerine karşı CNN mimarilerinin üzerinde direnç göstermektedir. Swin Transformer'ın kayan pencere attention mekanizması, $O(n^2)$ karmaşıklığını $O(n)$ düzeyine indirerek yüksek çözünürlüklü görüntülerde pratik kullanımı mümkün kılmaktadır.

EfficientViT ve MobileViT gibi hibrit CNN-Transformer mimarileri, ViT'in 86 milyon parametresini 10-20 milyon seviyesine indirerek edge uygulamalara uygun boyutlarda yüksek doğruluk sunmaktadır.

```
# insightface + FAISS: üretim entegrasyonu
from insightface.app import FaceAnalysis
import numpy as np, faiss

class FaceSystem:
    def __init__(self, gallery_path):
        self.app = FaceAnalysis(name="buffalo_l",
            providers=["CUDAExecutionProvider"])
        self.app.prepare(ctx_id=0, det_size=(640,640))
        data = np.load(gallery_path)
```

```
        embs = data["embeddings"]  # (N,512)
        self.ids = data["ids"]
        self.index = faiss.IndexFlatIP(512)
        self.index.add(embs)

    def identify(self, frame, thr=0.60):
        results = []
        for face in self.app.get(frame):
            emb = face.embedding
            emb = emb / np.linalg.norm(emb)
            sc, idx = self.index.search(
                emb.reshape(1,-1), 1)
            results.append({
                "bbox":  face.bbox.tolist(),
                "id":    self.ids[int(idx[0,0])],
                "score": float(sc[0,0]),
                "ok":    float(sc[0,0]) >= thr,
            })
        return results
```

11.2 Federe Öğrenme ile Mahremiyet Koruyucu Eğitim

Federe öğrenme (Federated Learning — FL), ham eğitim verilerinin merkezi bir sunucuya iletilmesi gerekmeksizin birden fazla tarafın ortak model eğitimine katılabildiği dağıtık bir paradigmadır. Yüz tanıma bağlamında birden fazla banka şubesi veya hastane, biyometrik verilerini paylaşmadan ortak bir gömme modeli eğitebilir. FedAvg algoritmasında (McMahan ve ark., 2017) her katılımcı yerel SGD adımları çalıştırır; merkezi aggregator ağırlık ortalamalarını alarak global modeli günceller.

FL'nin pratik zorlukları: veri heterojenliği (non-IID dağılım problemi), iletişim yükü ve model inversiyonu saldırıları (gradyanlardan orijinal verinin yeniden oluşturulabilmesi). Diferansiyel gizlilik (DP-SGD) — gradyanlara Gaussian gürültü eklenmesi — bu riski ε-DP garantisiyle matematiksel olarak sınırlandırır.

11.3 Deepfake Tespiti ve Karşı Önlemler

GAN ve diffusion model tabanlı deepfake üretim araçlarının demokratikleşmesi yüz tanıma sistemlerine yönelik yeni bir tehdit sınıfı oluşturmaktadır. Tespit yöntemleri: frekans bölgesi analizi (GAN'lar piksel spektrumunda karakteristik izler bırakır), zamansal tutarsızlık tespiti (temporal artefaktlar) ve rPPG (remote photoplethysmography — video'dan nabız sinyali çıkarılması).

Deepfake Yöntemi	Tespit Yaklaşımı	Başarı Oranı	Kısıt
Face2Face (landmark)	Frekans analizi	%96.2	Sıkıştırma sonrası düşer
DeepFaceLab (enc-dec)	Temporal tutarsızlık	%91.7	Kısa kliplerde zayıf
Diffusion model swap	rPPG sinyal analizi	%84.3	Hesaplama maliyeti yüksek
Neural radiance field	Birleşik dedektör	%78.5	Aktif araştırma alanı

Tablo 5. Deepfake üretim yöntemleri ve karşı tespit yaklaşımlarının tahmini başarı oranları (FaceForensics++ benchmark).

www.burakozgun.com

12. TARTIŞMA VE SONUÇLAR

12.1 Teknolojik Olgunluk ve Pipeline Gecikme Bütçesi

Kontrollü aydınlatma ve kooperatif senaryolarda ArcFace tabanlı sistemler %99.7 üzerinde doğruluk sağlamakta; Jetson Nano edge cihazında TensorRT INT8 ile 22 ms gecikme gerçek zamanlı işlemi mümkün kılmaktadır. 30 fps hedefi için toplam 33 ms bütçenin dağılımı: video çözme 2 ms, ön işleme 1 ms, yüz tespiti 5 ms, hizalama 1 ms, ArcFace embedding 8 ms, FAISS araması 1 ms, ağ gecikmesi 4 ms — toplam 22 ms (T4 GPU).

Pipeline Adımı	T4 GPU (ms)	Jetson Nano (ms)	Optimizasyon
Video çözme (H.264)	2	2	HW decode
Ön işleme + normalize	1	1	NumPy vektörize
Yüz tespiti (SCRFD)	5	8	TensorRT FP16
Yüz hizalama (Affin)	1	1	OpenCV C++
ArcFace embedding	8	22	TensorRT INT8
FAISS kimlik araması	1	1	IVF-PQ
Ağ gecikmesi	4	0	Edge offload
TOPLAM	22	35	T4: 30 FPS aşılır

Tablo 6. Gerçek zamanlı yüz tanıma pipeline gecikme bütçesi analizi. T4 GPU mimarisi 30 FPS hedefini karşılar.

12.2 Sınırlamalar ve Gelecek Yönelimler

Açık küme tanıma (open-set recognition), deepfake saldırıları ve aşırı poz değişimleri karşısında sistem hâlâ aktif araştırma konusu olmaya devam etmektedir. Hibrit mimari — edge'de tespit, merkezi GPU'da kimlik yönetimi — çoğu endüstriyel uygulamada en uygun dengeyi sunar. Gelecekteki araştırmaların; homomorfik şifreleme tabanlı gizlilik koruyucu çıkarım, demografik adalet kısıtlı eğitim yöntemleri ve deepfake savunması üzerinde yoğunlaşması beklenmektedir.

12.3 Sonuç

Bu çalışma; MTCNN/SCRFD yüz tespitinden ArcFace gömme çıkarımına, TensorRT optimizasyonundan FAISS tabanlı kimlik aramasına uzanan uçtan uca pipeline'ı; ViT tabanlı mimari yenilikler, federe öğrenme, deepfake tespiti ve etik-hukuki çerçeveyle birlikte kapsamlı biçimde ele almıştır. LFW'de %99.82 doğruluk ve 22 ms edge gecikme, teknolojinin hem doğruluk hem de gerçek zamanlı kısıtı karşıladığını kanıtlamaktadır. KVKK ve GDPR uyumluluğu, açık rıza mekanizmaları ve demografik yanlılık denetimi; teknolojinin toplumsal kabulü ve sürdürülebilirliği açısından teknik doğruluk kadar belirleyicidir.

www.bulakol.org

KAYNAKÇA

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. Proceedings of the IEEE/CVF CVPR, 4690–4699. https://doi.org/10.1109/CVPR.2019.00482

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. Proceedings of the IEEE CVPR, 815–823. https://doi.org/10.1109/CVPR.2015.7298682

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., & Liu, W. (2018). CosFace: Large margin cosine loss for deep face recognition. Proceedings of the IEEE/CVF CVPR, 5265–5274. https://doi.org/10.1109/CVPR.2018.00552

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. Proceedings of the IEEE CVPR, 1701–1708. https://doi.org/10.1109/CVPR.2014.220

Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). RetinaFace: Single-shot multi-level face localisation in the wild. Proceedings of the IEEE/CVF CVPR, 5203–5212. https://doi.org/10.1109/CVPR42600.2020.00524

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE CVPR, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE CVPR, 770–778. https://doi.org/10.1109/CVPR.2016.90

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. ICLR. https://arxiv.org/abs/2010.11929

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of the 1st Conference on Fairness, Accountability and Transparency, 81, 77–91.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the IEEE CVPR, 1, 511–518. https://doi.org/10.1109/CVPR.2001.990517

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. NIPS Deep Learning Workshop. arXiv:1503.02531.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of AISTATS, 54, 1273–1282.

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. Proceedings of the IEEE/CVF ICCV. https://doi.org/10.1109/ICCV.2019.00009

Liu, Y., Jourabloo, A., & Liu, X. (2018). Learning deep models for face anti-spoofing. Proceedings of the IEEE/CVF CVPR, 389–398. https://doi.org/10.1109/CVPR.2018.00048

Grother, P., Ngan, M., & Hanaoka, K. (2019). Face recognition vendor testing (FRVT) Part 3: Demographic effects. NIST IR 8280. https://doi.org/10.6028/NIST.IR.8280

www.bulkoglan.com

European Parliament and Council. (2024). Regulation on a European approach for artificial intelligence (EU AI Act). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

Kişisel Verileri Koruma Kurumu. (2016). 6698 sayılı Kişisel Verilerin Korunması Kanunu. Resmi Gazete, 29677. https://www.mevzuat.gov.tr/MevzuatMetin/1.5.6698.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

www.burakolgun.com