T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BÜYÜK ÖLÇEKLİ YÜZ TANIMA SİSTEMLERİNDE ARAMA UZAYININ KÜÇÜLTÜLMESİ AMACIYLA VERİLERİN TEMSİL BENZERLİKLERİNE GÖRE KÜMELENDİRİLMESİ

Özgür ÖZKAN

DANIŞMAN
Doç. Dr. Selçuk SEVGEN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

İSTANBUL-2020

Bu çalışma 13.01.2020 Tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı Yüksek Lisans Tezi olarak kabul edilmiştir.

TEZ JÜRİSİ

Doç. Dr. Selçuk SEVGEN
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Rüya ŞAMLI
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Dr.Öğr.Üyesi Zeynep TURGUT
Haliç Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

İnsanlar hayatlarındaki önemli anların fotoğraflarını kendi çekememekte ve kendi çekmek yerine başka biri tarafından fotoğraf çekilmesi durumunda kalmaktadır. Hayattaki önemli anların fotoğraflarını farklı kişilerin çekmesi onları çekilen kişilerin cihazlarına dijital dosyayı ulaştıramama problemi oluşturmaktadır. Bu problem mezuniyet, düğün, maraton gibi kalabalık etkinliklerde daha da artmaktadır. Fotoğrafçıların bu tip etkinliklerde çektiği fotoğraflar katılımcılara ulaşamamaktadır. Bu sorunu çözmeye çalışırken çözülmesi gereken problemin yüz tanıma sistemleriyle fotoğraf ulaştırmak olduğu açıktır. Literatürde ve uygulamada öklit uzaklığı bazlı yakın komşu bulan hızlı sistemlerin hazır olarak kullanılamayışı daha iyi bir yöntem bulunabilmesi için araştırmaya kaynak olmuştur.

Bu araştırmayı gerçekleştirirken kaynaklarından yararlandığım şirket Reminis'e, araştırma sürecinde anlattıklarımı sıkılmayarak dinleyen ortaklarım Utku Gürçağ Borataç, İzzet Yücel Peynirci ve Volkan Sezer'e, beni destekleyen Hülya Özkan ve Ece Gözde Ergür'e teşekkür ederim. Son olarak sevgili danışmanım Doç.Dr. Selçuk SEVGEN'e ve tüm komite üyelerine teşekkür ederim. Hepinize desteğiniz için teşekkür ederim.

Ocak 2020 Özgür ÖZKAN

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **IV**
İÇİNDEKİLER ..... **V**
ŞEKİL LİSTESİ ..... **VI**
TABLO LİSTESİ ..... **VII**
SİMGE VE KISALTMA LİSTESİ ..... **VIII**
ÖZET ..... **IX**
SUMMARY ..... **XI**
1. GİRİŞ **1**
1.1 HEDEF VE AMAÇ 2
1.2 ZORLUKLAR 4
1.2.1 Hız *4*
1.2.2 Doğruluk *4*
1.2.3 Ölçeklenebilme *5*
1.2.4 Uygulanabilme – İşlem verimliliği *5*
1.2.5 Halka açık kapsayıcı gözetimli veri kümelerinin eksikliği *5*
2. GENEL KISIMLAR **6**
2.1 FOTOĞRAFLARDAKİ YÜZLERİN- TESPİTİ 6
2.1.1 Yönlü Gradyanların Histogramı Metodu *6*
2.1.2 Yüz Fotoğraflarının Karşıdan Çekilmiş Yüz Fotoğraflarına Dönüştürülmesi *7*
2.2. YÜZLERİN KARŞILAŞTIRILABİLİR ŞEKİLDE MATEMATİKSEL TEMSİLİ 8
3. MALZEME VE YÖNTEM **10**
4. BULGULAR **22**
4.1. PERFORMANS DEĞERLENDİRMESİ 26
5. TARTIŞMA VE SONUÇ **36**
KAYNAKLAR **40**
ÖZGEÇMİŞ **43**

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 Bir fotoğrafın HOG yöntemi ile gösterimi……7

Şekil 3.1 Yüz tespit edilen gerçekte yüz olmayan fotoğraflar……11

Şekil 3.2 2 boyutlu uzayda orjini kapsayabilen dairesel-hacimsel şekiller...……13

Şekil 3.3 2 boyutlu uzayda orjine yetişmeyen 4 nokta ……14

Şekil 3.4 3 boyutlu uzayda orjini kapsayan 6 dairesel-hacimsel şekil.……14

Şekil 3.5 3 boyutlu uzayda orjinde aynı yarıçapa sahip dairesel-hacimsel şekil ……..…….……15

Şekil 3.6 B'nin yakın komşusu A ve D'nin bulunabildiği senaryo…..……16

Şekil 3.7 B'nin yakın komşusu D'nin üst katmandaki A noktasıyla eşleşmediği için bulunamadığı senaryo..……16

Şekil 4.1 CMC Skoru Karşılaştırması……..25

Şekil 4.2 Lineer liste araması (sol) ve yeni yöntem (sağ) CMC grafiği...……..27

Şekil 4.3 Yeni yöntem(sol) CMC grafiği Lineer liste araması(sağ) tüm doğru pozitifler……..28

Şekil 4.4 IET Grafiği……....33

Şekil 4.5 Önerilen FNIR değeri ile DET Grafiği…….34

TABLO LİSTESİ

Sayfa No

Tablo 4.1 Kimlik tanımlamada galeri ve arama bilgisine göre problemlerin gruplandırılması…… 22

Tablo 4.2 Açık-Küme Kapalı-Küme problemi……. 23

Tablo 4.3 Tek senaryo için örnek sonuçlar…… 25

Tablo 4.4 T=0.60 eşiğinde arama sonuçları……. 28

Tablo 4.5 FPIR@0.2 değerinde arama sonuçları…… 30

Tablo 4.6 FPIR@0.02 değerinde arama sonuçları…… 31

Tablo 4.7 FPIR@0.002 değerinde arama sonuçları…… 32

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar **Açıklama**

d : Bir verideki farklı boyut sayısı

CMC : IJB-B veri kümesinde yer alan performans metriği kümülatif eşleşme puanı

IET : Kimlik bulma hatasında hatayı düşürmek için doğru sonuçtan feragat etme puanı

IJB-B : IARPA Janus Benchmark isminde bir yüz veri kümesi

1:N : 1 yüzle N tane yüz fotoğrafında doğru kimliği arama problemlerine verilen genel ad

HOG : Yüz fotoğraflarındaki karanlıktan açığa geçişi temsil etmek için kullanılan özellik ismi

Resnet : Üst katmandaki artık verilerin derin katmanlara tekrar girdi olarak kullanıldığı nöral ağ

S1 : IJB-B kümesindeki 1 numaralı galeri

S2 : IJB-B kümesindeki 2 numaralı galeri

FNIR :Yanlış negatif kimliklendirme oranı

FPIR :Yanlış pozitif kimliklendirme oranı

Rank-1 :Tahmin listesinde en yüksek ihtimalle bulunmuş sonuçta doğru pozitif bulunma oranı

Rank-10 :Tahmin listesinde en yüksek 10 ihtimalle bulunmuş sonuçta doğru pozitif bulunma oranı

Rank-N Total(R-NT):Tahmin listesinin toplamında bulunan tüm doğru pozitiflerin tahmin listesindeki sırasının tahmin listesi boyutuna oranı

VGG,LFW,Face-Scrub: Yüz veri kümeleri

ÖZET

YÜKSEK LİSANS TEZİ

BÜYÜK ÖLÇEKLİ YÜZ TANIMA SİSTEMLERİNDE ARAMA UZAYININ KÜÇÜLTÜLMESİ AMACIYLA VERİLERİN TEMSİL BENZERLİKLERİNE GÖRE KÜMELENDİRİLMESİ

Özgür ÖZKAN

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Doç. Dr. Selçuk SEVGEN

Yüz tanıma, kontrollü şartlarda insan başarımına yakın ve ondan başarılı sonuçlar vermektedir. Günümüzde yüz tanımadaki yüksek başarım kontrollü şartlarda sağlanan bire bir karşılaştırmalarda elde edilmiştir. Kontrolsüz şartlarda oluşturulmuş listelerde yüz tanıma veya özdeşleştirme yapma problemini çözme başarımı henüz bire bir karşılaştırma problemini çözme başarımına yaklaşamamıştır. Bu probleme açık küme ismi verilmektedir. Bu çalışmada açık küme listelerinde yapılan kimlik bulma problemine çözüm üretilmesi amacıyla yaklaşık yakın komşu aramasına yeni bir yaklaşım getirilmiş ve lineer liste aramasıyla arasındaki performans farkı karşılaştırılmıştır. Çalışma süresi ve karmaşıklığı bakımından daha avantajlı sonuçlar ürettiği gösterilen çözüm herhangi bir yüz doğrulama yöntemini yüz kimlik bulma yöntemi olarak kullanmaya imkan tanımakta olup, herhangi bir yüz tanıma yöntemine girdi oluşturan listeyi organize edip, böl-birleştir yöntemiyle listeyi ve fonksiyonu spesifik listelere ve adımlara bölerek arama uzayını üssel olarak küçültmektedir. Açık küme problemlerinde yüz tanımanın temsil benzerliklerinin dağılımının avantajından yararlanan bu yöntem, herhangi bir bire bir yüz tanıma yöntemini listede kimlik bulma yöntemine dönüştürmektedir. Açık küme listesinin sürekli büyüdüğü, sık güncellendiği durumlarda karşılaşılan problemleri çözmesi bakımından

özgün bir yöntemdir. Çalışma kapsamında lineer liste arama yöntemiyle sunulan özgün yöntem IJB-B yüz kimliği tanıma veri kümesinde performans ve çeşitli ölçeklerde karşılaştırılmıştır.

Ocak 2020, 54 sayfa.

Anahtar kelimeler: Yüz, Tanıma, Açık, Küme, Yapay, Zeka

SUMMARY

[M.Sc. THESIS]

[Representational Similarity Clustering For Search Space Reduction of Large Scale Face Recognition]

[Özgür ÖZKAN]

Istanbul University-Cerrahpasa

Institute of Graduate Studies

[Department of Computer Engineering]

Supervisor : [Assoc. Prof. Dr.] [Selçuk SEVGEN]

[Today, face recognition shows human-like performance in constrained settings. Although the face verification problem is almost solved, face identification problem which occurs when a possible match is searched against a face list is still lacking in unconstrained environments. This problem is called an open-set search. In this novel work, a new approach to approximate neighbor search is introduced on the open-set facial identification problem and it is compared to linear search. In terms of search time complexity and distributivity to scale, the new method that is proposed transforms any face verification algorithm to a face identification algorithm by doing a divide and conquer on face input list to prune the search space exponentialy and it slightly changes the benchmark scores when compared to the linear search version because of the nature of approximate neighbor search. As the results show this work's novel approach is favorable when the dataset changes faster dynamically and nondeterministically because it requires very minimal preprocessing when compared to other approximate neighbor search algorithms and data structures. The results of IJB-B's 1:N open set protocol are used to compare the performance between the new approach and the linear list scan.]

January 2020, [54] pages.

Keywords: [Face recognition,open-set face identification,approximate neighbor search]

1. GİRİŞ

Günümüzde yüz tanıma sistemleri hem bilimsel hem pratik açıdan oldukça gelişmiştir. Güvenlik, kimlik tanıma ve doğrulama, yetkilendirme, personel ve öğrenci yoklama, ödeme, fotoğraf ulaştırma, sosyal medya gibi alanlarda kullanılmakta olan yüz tanıma sistemleri hayatımızın birçok alanına entegre olmasıyla vazgeçilmezdir.

Yüz tanıma problemi doğrulama ve kimlik bulma şeklinde ikiye ayrılmaktadır. Doğrulama iki yüzün aynı kişiye ait olup olmadığının cevabını verme amaçlı yapılırken, kimlik bulma bir liste içinde aranan kimliğin hangisi olduğunun cevabını vermeye çalışır. Kimlik bulma problemlerinde aranılan kimlik, arama listesinde mevcut değilse ve yüz tanıma yönteminin öğrenim kümesinde hiç bulunmamış bir yüzle arama yapılıyorsa bu biçimde bir problem bağlamdan bağımsız bir açık küme problemi olarak isimlendirilmektedir [1].

Açık küme problemlerinde bire bir yüz doğrulama problemini çözme amacıyla geliştirilmiş yöntemler kullanıldığı için başarım oranları diğer senaryolara oranla çok daha düşüktür. Ayrıca açık küme probleminin başarım ölçütleri de yüz doğrulama yöntemlerinden farklıdır. Yüz doğrulama yöntemi, yayınlanmış veri kümelerinde kaç tane bire bir karşılaştırmanın yöntem tarafından doğru tahmin edildiği şeklinde bir başarı ölçütüne sahiptir. Kimlik bulma yöntemi ise listede aranan kimliğin hangileri olabileceğini tahmin eder ve genelde ilk 10 ve 20 sonuçta doğru sonucu kaçıncı sırada tahmin edilebildiğiyle hesaplanır. Tek kimlik tahminlerinde başarım yarı yarıya düşmekle birlikte arama listesinde bulunmayan bir kimlikle arandığında algoritmanın sonuç döndürmemesi daha yüksek bir başarıdır.

Gunther (2017) akademik gelişmelerin çoğunluğunun kapalı küme problemine odaklandığını belirtmiş, ve açık küme sorununa kısmen daha az odaklanıldığını belirtmiştir [2]. Yüz tanıma sistemlerinde sorgu hızıyla ilgili ise oldukça az çalışma mevcuttur [3]. Açık küme yüz tanıma sistemlerinde tüm dünyada kabul gören bir performans protokolü mevcut değildir [4]. Büyük çaplı arşivlerde çalışan yüz tanıma algoritmalarını doğru karşılaştırmak için performans karşılaştırmayı sağlayacak yöntemlerden biri Nech (2017) tarafından sunulmuştur [5].

Kapalı küme yüz tanıma problemi aranacak listede aranacak kişinin fotoğrafının kesinlikle bulunduğunun bilinmesiyle tanımlanmaktadır. Bu konuda yüz tanıma oldukça başarılı sonuçlar vermektedir. Diğer konu olan açık küme yüz tanıma probleminde ise aranacak kişinin aranacak

listede olup olmadığının bilinmemesi en büyük sorunlardan biridir. Bu tür problemlerde listede bulunmayan bir yüz araması için eşleşme tahmininin hiç sonuç vermemesi gerekir. Yanlış sonuçları vermeme gereksinimi yüz tanıma algoritmalarının performansını düşürmektedir. Aranan yüzün gerçekte kendisi olmayan yüzlerle karşılaştırma sayısının artması yüz tanıma algoritmalarının hatalı kararlar vermesine yol açmaktadır. Çalışmanın amacı arama listesinin boyutunun çok yüksek olduğu durumlarda hem arama hızını artırmak hem de hata oranını düşürebilecek bir yöntem oluşturup bu yöntemin akademik yüz tanıma veri kümelerinde uygulanarak yarışma sonuçlarıyla karşılaştırılmasıdır. Çalışma açık küme yüz tanıma probleminin çözüm başarımını hem hız hem de hata oranı değişkenlerine bağlı olarak artırmayı amaçlamaktadır.

Literatürdeki çalışmalarda önerilen yeni yöntemler genelde arama listesindeki her yüzle tek tek karşılaştırma yapmasına dayanmaktadır. Araştırma yöntemleri kendi aralarında karşılaştırıldığında farklı veri kümesi, farklı öznitelikler, farklı öğrenim metotları kullanmaktadır. Bu çalışmada kullanılacak yöntem bir kabule bağlı olarak arama listesini filtrelemektir. Arama listesinin filtrelendiği çalışmalardan biri Das (2011) tarafından yapılmıştır [6]. Bu çalışma biyometrik verilerde öğrenim problemlerinde öğrenilen gizli özelliklerden yararlanarak arama listesinin boyutunu küçültmeyi amaçlamaktadır. Çalışma, akademik çalışmalarda kullanılan yaygın açık küme yüz tanıma veri kümesi olan IARPA JANUS BENCHMARK veri kümesi serisinin IJB-B veri kümesi üzerinde gerçeklenmiştir [7][8].

Çalışma kapsamında yöntemi karşılaştırma amacıyla Maze (2018) tarafından yayınlanan veri kümesi kullanılarak o çalışmada belirtilen 1:N protokolünde eşik değer bazlı FPIR, FNIR metrikleri ve sıralama bazlı CMC metriği kullanılmıştır [9]. IJB-B üzerinde 1:N protokolü listede aramayı atomik olarak değerlendirmektedir ve sınırlama koymamaktadır [7].

1.1 HEDEF VE AMAÇ

Yüz tanıma sistemlerinin gerçek dünya problemlerine uygulanmasında sık karşılaşılan sorunlardan biri gerçek dünya problemlerinde eşleştirilecek veya kimlik tespiti yapılacak unsurların farklı bağlamlara ait olmasıdır. Arama işlemi farklı bir bağlamdan bir bilgi ile başka bir bilinmeyen bağlamdan bilgi çıkarımı fonksiyonudur. Birbirinden bağımsız bağlamlarda kimlik aramasını başarılı şekilde yapabilmek için bağlamlar arası bilgilerin öğrenim veri kümesinde bulunması gerekmektedir. Aranan yüzün yüz doğrulama yönteminin

oluşturulmasında kullanılan öğrenim kümesinde olmaması, aranan yüze ait kimliğin başka bir yüzünün aranacak listede olmaması, arama yapılan galeri yüzlerinin kullanılan doğrulama yönteminin öğrenim kümesinde bulunmaması, arama yapılan galerinin büyük ölçekli olması, kullanılan doğrulama yönteminin öğrenim kümesinde yer alan kimliklerin genetik olarak arama ve galeri kimliklerine uzak olması, galeride genetik özellikleri birbirine benzer kişilerden çok sayıda bulunması, yüzün bulunduğu fotoğrafı çeken kameranın kullandığı fotoğraf düzeltme işlemleri , gerçek dünya problemlerinde kimliklerin yüz özelliklerini kapatan saç ve şapka gibi unsurlar, ışık ve renklilik seviyesi birbirinden oldukça değişik fotoğraflar açık küme 1:N kimlik araması probleminin başarımını oldukça düşüren başlıca problemlerdendir [10].

Açık küme 1:N kimlik araması probleminin çözümü güvenlik alanında kalabalıklar içinden suçluyu tespit etmeye, çevrimiçi ağda ve belirli bir veri tabanındaki milyonlarca yüz içerisinden kendi yüzümüzün bulunduğu fotoğrafları bulmaya, yüzle ödeme sistemlerine, büyük marketlerin yüz tanıma bazlı sadakat programları yapabilmesine, özelleştirilmiş reklamlara, göçmen büronun ihlalcileri tespit etmesine kadar birçok alanda uygulanmaya çalışılmaktadır.

Açık küme 1:N kimlik araması bir benzerlik araması olmasıyla öneri sistemlerine çok benzemektedir. Bu çalışmada, açık küme 1:N kimlik araması probleminde ve 1:N kimlik aramasının yanında aynı kimliğe ait tüm diğer aday yüzlerin yüksek başarım ile bulunamaması probleminin çözümü için bir yaklaşık yakın komşu araması grubunda olabilecek bir veri yapısı ve arama algoritması önerilmektedir.

Yaklaşık yakın komşu araması, çok sayıda ve çok boyutlu özniteliğe sahip veri kümesinde benzer verilerin bulunması için geliştirilen çalışmaların genel ismidir. Milyonlarca veride yeni bir veriyi tek tek kümedeki her bir veri ile karşılaştırmak zaman alan bir işlemdir. Arama süresinin küme büyüklüğüyle lineer şekilde artmasına sebep olan tek tek karşılaştırma yöntemindense yaklaşık yakın komşu araması algoritmaları kesin komşuları getirme kuralını ortadan kaldırarak her bir veriyle karşılaştırma yapmadan yakın komşuları belli kayıplarla getirmeyi öngörmektedir. Kesinlik özelliğinin kaybı kapalı küme liste aramalarında herhangi bir algoritmanın başarım ihtimalini düşürecek olsa da açık küme liste aramalarında yanlış pozitif, doğru negatif sonuçları azaltacağı öngörülmüştür. Oluşturulan arama listesi galerilerinde aranan kimliğe ait yüzlerin sayısı galerideki o kimliğe ait olmayan yüzlerin yanında çok düşük bir oranda bulunduğu için kesin yakın komşu aramalarında komşunun yanlış komşu olma ihtimali çok yüksektir.

Açık küme 1:N kimlik araması protokollerinde ve halka açık veri kümesi çalışmalarında kimliğin birden çok yüzünün galeride bulunması söz konusudur. Buna rağmen algoritma başarımlarını karşılaştıran protokollerin öne sürdüğü başarım metrikleri listedeki doğru sonucu en yüksek ihtimalle tespit etmekten öteye giden metrikler sunmamışlardır. Bu çalışma kapsamında açık küme 1:N kimlik araması protokolü olarak yeni bir metrik önerilecektir. Günümüzde halka açık veri kümelerinde önerilen metrikler gerçek dünyada sadece kimliğe ait bir yüzün bulunmasının yeterli olduğunu öne sürmektedir. Fakat bu gerçek dünya uygulamalarında şüpheli kimliğin en yeni veya tüm görünüş biçimlerinin gözden kaçırılmasına sebep olmaktadır. O yüzden algoritmaların bir kimliğe ait tüm yüzleri ne kadar başarılı getirdiği bu gerçek dünya uygulamalarındaki başarımını belirlemelidir. Galeride şüpheli aranan kimliğin birden çok yüzünün olması daha gerçekçidir. Sadece birini en yüksek olasılıkla getirmesi yeterli değildir. Bu çalışma kapsamında kimliğin galerideki tüm yüzlerinin tahmin listesindeki sıralarının kullanılarak hesaplandığı yeni bir başarı metriği önerilecektir. Bu başarı metriği bir kimliğin galerideki tüm fotoğraflarının ne kadar yüksek olasılıkla tahmin edilebileceğinin hesabıdır.

Lineer benzerlik araması algoritması yüz tanıma sistemlerinde en yakın komşuları %100 kesinlikte bulmaktadır. Algoritmanın uygulanışı büyük ölçekli yüz listelerinde çok vakit almaktadır. Bu çalışma kapsamında, lineer benzerlik aramasının çalışma hızı önerilen çözümün arama hızıyla karşılaştırılacaktır.

1.2 ZORLUKLAR

Yüz tanıma problemlerinin çözümündeki zorluklardan bazıları bu bölümde açıklanmıştır.

1.2.1 Hız

Milyonlarca kimliğin çok sayıda yüz fotoğrafının olduğu bir galeride tahmin listesini elde etmek için gereken süre yüksektir. Bu sorunu aşmak amacıyla verileri indeksleme yöntemleri kullanılmaktadır. Kafai (2014) Hash yöntemi kullanarak verileri indeksleyip arama süresini kısaltmıştır [11].

1.2.2 Doğruluk

Galerideki fotoğraf sayılarının yüksek olması, fotoğraflardaki bulanıklık, kişinin yüz bilgilerini kapayan unsurların fotoda yer alması gibi unsurlar doğruluğu etkilemektedir. Park (2010) kişiye özel akne, leke gibi yumuşak biyometrik verilerin doğruluğu artırdığını göstermiştir [12].

1.2.3 Ölçeklenebilme

Fotoğraf sayılarının sürekli arttığı galerilerde yeni verilerin veri madenciliği yapılarak aramaya hazır hale getirilmesi de başlı başına bir problemdir. Facebook, 2013'te günde 250 milyon yeni fotoğraf yüklendiğini belirtmiştir. Kısıtların olmadığı fotoğraflarda yüz tanıma işleminin büyük galerilerde gerçekleştirilmesi, yüz tanımanın günümüzde gelişmiş olmasına rağmen hala düşük başarımlı sonuçlar vermesine neden olmaktadır [13].

1.2.4 Uygulanabilme – İşlem verimliliği

Yüz tanıma algoritmaları yüz bilgilerinin matematiksel temsilleriyle işlemler yaptığından dolayı matris yoğunluklu işlemler yoğundur. Matris yoğunluklu işlemler grafik işlemcilerde daha hızlı işlenmektedir. Grafik işlemcili bilgisayarların büyük ölçekli yüz tanıma sorunlarının çözümü için altyapı sağlayıcılardan dinamik kiralanması gerekmektedir. Dinamik kiralama yapılabilecek grafik işlemcili bilgisayarların ulaşılabilirliği sadece işlemci bulunan bilgisayarlara göre daha azdır. Bunun yanında literatürdeki çalışmaların çoğu grafik işlemcili tek bilgisayarda çalışan algoritmalar üzerinedir. Literatürde önerilen yöntemler dağıtık sistemlerde çalışabilecek algoritmaların çalışmasına imkân veren veri yapıları içermemektedir. Çalışmalar bir diske doğrudan erişimi olan sistemlerde çok hızlı çalışırken dağıtık sisteme geçildiğinde ağ kullanım ihtiyacından doğan hız düşüşleri literatürdeki algoritmaların dağıtık sistemlerde avantajını kaybetmesine yol açmaktadır.

1.2.5 Halka açık kapsayıcı gözetimli veri kümelerinin eksikliği

Algoritmalar yüz verilerinin matematiksel temsilini oluştururken gözetimli veri setleri kullanılarak eğitilmektedir. Günümüzde veri setlerinin oluşturulması zaman alıcı ve detaylı bir süreçtir. Kapsayıcılığı sağlamak için ise her coğrafyadan yüzlerin kullanılması gerekmektedir.

2. GENEL KISIMLAR

Bu bölümde, çalışma kapsamında kullanılan bilgisayarla görü, derin öğrenme ve görüntü işleme adımları açıklanmıştır.

2.1 FOTOĞRAFLARDAKİ YÜZLERİN- TESPİTİ

Yüz tespiti, fotoğraflarda yer alan insan yüzlerinin bulunduğu piksellerin çeşitli algoritmalar yardımıyla tespit edilmesidir. Tespit sonrasında yüzün bulunduğu pikselleri kapsayan bir sınırlayıcı kutucuk kullanılmaktadır. Bu kutucuk yüzün bulunduğu fotoğrafta yüzün olduğu yeri belirtmek için kullanılmaktadır. Tıpkı fotoğraf gibi yüksekliği, genişliği olan bu kutucuğun başlangıç noktası ana fotoğraftaki belirli yükseklik ve genişlik değeriyle tanımlanır.

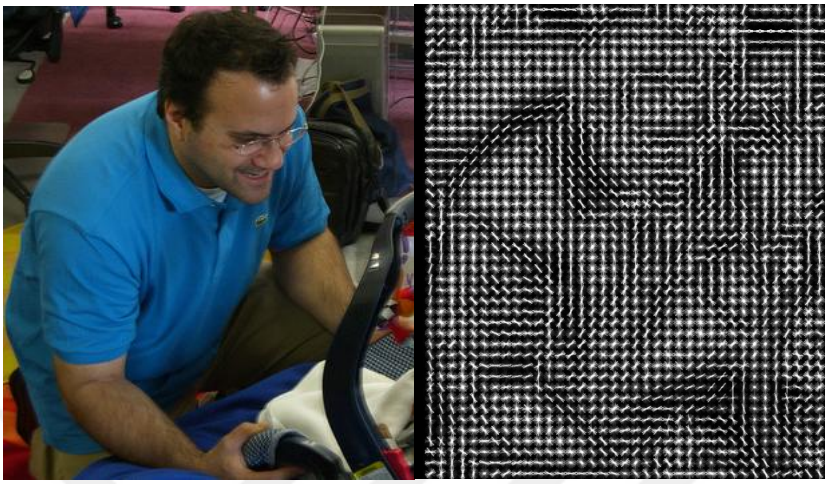
Yüz tespiti 2001 yılında ana akım literatür çalışmaları tarafından ilgi görmüştür. Paul (2001) yüz tanıma için sınıflandırıcıyı saniyede 15 karede çalışacak şekle getirmiştir [14]. Çalışma kapsamında yüz tespitinde Yönlü Gradyanların Histogramı (Histogram of Oriented Gradients-HOG) tanımlayıcıları kullanılmıştır. HOG tanımlayıcıları Gabor-EBGM yöntemine göre ışık, yön değişimleri gibi durumlarda daha iyi sonuç vermektedir. [15]

Çalışma kapsamında kullanılacak HOG metodu 2005 yılında ki bir çalışmaya dayanmaktadır. [16]

2.1.1 Yönlü Gradyanların Histogramı Metodu

Görüntü işleme metotlarında genellikle pikseller komşu piksel değerleriyle karşılaştırılmaktadır. HOG metodu, her bir piksel için komşu piksellerle karşılaştırıldığında ne kadar karanlık bir değere sahip olduğu bulunduktan sonra değerlendirilen piksellerden en parlak değerli olandan en karanlık değerli olana doğru bir ok çizilmesiyle fotoğraf piksellerinin her birinin bu oklarla temsil edilmesi alt yöntemini kullanmaktadır. Bu oklara gradyan adı verilmektedir. Gradyanlara dönüştürülmüş fotoğraf bir fotoğraftaki piksellerde parlak ve karanlık piksel değerlerinin akışını temsil etmektedir. Piksel değerleri yerine bu yöntemin kullanılmasının nedeni çok karanlık veya çok parlak fotoğraflarda yüz tanıma yapılmasının ayrı iki problem teşkil etmesidir. Kullanılan yöntemle bu iki problem çözülmesi gereken tek bir probleme dönüşmektedir. Her bir pikselin gradyan oku ile temsili çok sayıda detay içermektedir. Bunun yerine 16x16 boyutunda pikselleri kümeleyerek karanlıktan, parlak yöne

en güçlü geçiş kabulü bulunmaktadır. Bu yöntemle oluşturulan fotoğraf gösterimi Şekil 2.1'de görülebilir.



Şekil 2.1 Bir fotoğrafın HOG yöntemi ile gösterimi [16].

HOG yöntemi ile dönüştürülen fotoğrafta yüz bulunabilmesi için çok sayıda yüz olan bölgeleri işaretlenmiş fotoğraf HOG yöntemi ile dönüştürüldükten sonra bir örüntü üretilmektedir. Bu örüntünün varlığının başka HOG yöntemi ile dönüştürülmüş fotoğrafta aranması yüzün bulunmasını sağlamaktadır.

2.1.2 Yüz Fotoğraflarının Karşıdan Çekilmiş Yüz Fotoğraflarına Dönüştürülmesi

Fotoğraflarda yer alan poz değişikliklerinde yüz doğrudan kameranın karşısında olmayabileceği için HOG metodu tek başına yandan görülen yüzlerin tespitinde yeterli olmamaktadır. Bu problemin çözümü için yandan olan yüzün karşıdan çekildiği haline ihtiyaç duyulmaktadır. Bunu yapmak için ise fotoğraftaki yüzün bükülmesi gerekir. Bunu yapmak için milisaniyeler içinde yüzü bükebilmek için rasgele bir grup regresyon ağacının kullanıldığı ERT algoritması kullanılmıştır [17].

Yöntem yüzü bükebilmek için önce yüzde bulunan belirli noktaların tahminini yapmaktadır. 68 sabit noktanın tahminini yapan bu algoritma gözlerin, kaşların bulunduğu yerleri, dudakların yerlerini işaretlemektedir. Bu işlemden sonra karşıdan görünen bir yüzün işaretleri nereye geliyorsa o noktalara denk getirebilmek için diğer yüzün fotoğrafı afin dönüşüm işlemiyle bükülmektedir.

2.2. YÜZLERİN KARŞILAŞTIRILABİLİR ŞEKİLDE MATEMATİKSEL TEMSİLİ

Yüz tanımanın gerçekleştirilebilmesi için bir kişiye ait yüzlerin aynı kişiye ait olduğunu tahmin eden, farklı kişilerin yüzlerini de birbirinden farklılaştırabilen bir yöntem gerekmektedir. Bir insan hangi özelliklerin özgün kimlikleri tespit edebileceğini düşünerek tahmin edebilir fakat literatürdeki çalışmalar hangi özelliklerin önemli olduğunu insanların başarılı şekilde tahmin edemediğini göstermektedir. Bu sebeple derin öğrenme yöntemleri kullanılarak yüzlerin hangi kimliğe ait olduğuna karar veren özelliklere bilgisayarların veri ile karar vermesi daha iyi sonuçlar vermektedir.

Yüz kimliklerini tahmin etmek için içinde yüz olduğu bilinen fotoğraflar önce matematiksel bir vektöre çevrilmektedir. Bu matematiksel vektöre dönüştürme işlemi için özelleştirilmiş bir ResNet sinir ağı kullanılmıştır.

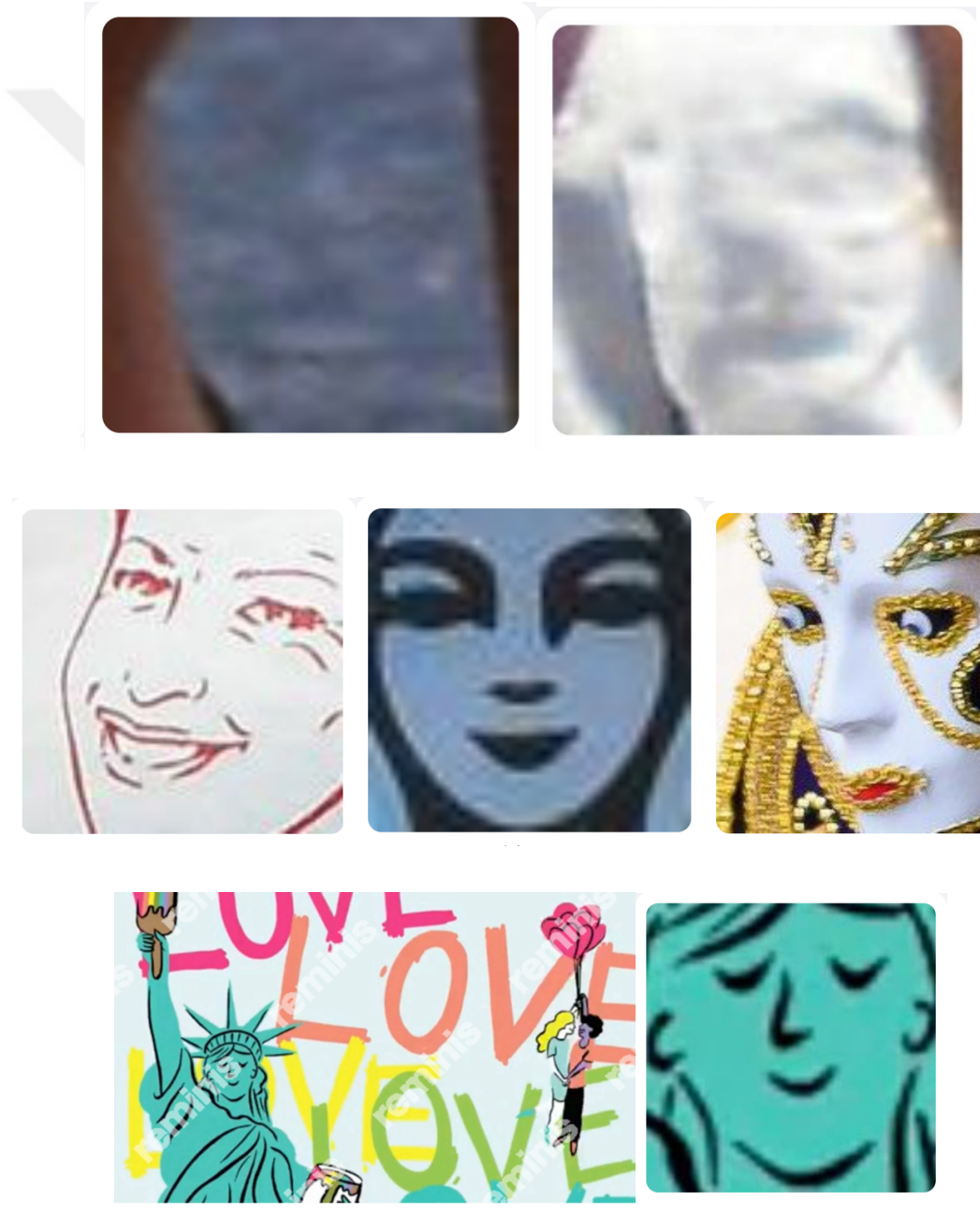
ResNet, artık değerli sinir ağ demektir. Evrişimli sinir ağlarının derinliklerinden kaynaklı olarak yakınsamaya başladığı zaman bozulma problemini çözmek için özelleştirilerek oluşturulmuştur. Evrişimli sinir ağlarının katmanları arasına kısayollar eklenerek derinleştikçe oluşan bozulmalar önlenmektedir. Çalışmada yüzler 29 katmanlı ResNet derin öğrenme modeli kullanılarak 128 boyutlu vektöre dönüştürülmektedir. ResNet derin öğrenme modeli ResNet-34 ağındaki bazı katmanların kaldırılıp, katmandaki filtrelerin yarıya düşürülmesiyle elde edilmiştir ve 3 milyon yüzle eğitilmiştir. 3 milyon yüz Face Scrub [18] ve VGG [19] veri kümeleri kullanılarak elde edilmiştir. Veri kümesinde 7485 farklı kimlik bulunmaktadır. Bu model LFW [20] değerlendirme protokolünde 1:1 karşılaştırmada %99.38 başarım elde etmiştir.

ResNet eğitilirken birden çok katmanlı ağ içinde fotoğraftaki pikseller bir matris şeklinde bu katmanlarda belirli bir fonksiyonda işlem görerek bir diğer katmana girdi olarak gönderilir. Katmanların en sonundan çıkan vektörler kendi aralarında kosinüs veya öklit uzaklıkları hesaplanarak karşılaştırılır aynı kimliğe ait yüzlerin vektör uzaklıklarının birbirlerine yaklaşması ve farklı kimliklere ait vektörlerin birbirinden uzaklaşması gerektiğinden fonksiyon içindeki ağırlık parametreleri sonuçları bu şekilde güncelleyecek yönde değiştirilir. Fonksiyonlarda yer alan ağırlıklardaki bu değişim fonksiyonların veriler sayesinde öğrenmesini sağlar. Yeterli veriyle eğitilmiş çok katmanlı iki yüzü karşılaştıran ResNet daha sonrasında daha önce öğretilen verilerde olmasına bile gerek olmayan farklı bir yüz verisini girdi olarak alıp tekrar bir vektör üretir. Artık yeterince öğrendiği için aynı kimliğe ait yüzlerin vektörlerinin

birbirleriyle kosinüs veya öklit uzaklığı ile karşılaştırımı birbirine yakın çıkarken aynı kimliğe ait olmayanlar birbirinden uzak çıkacaktır. Sonrasında bu matematiksel vektörler kullanılarak iki yüz fotoğrafının aynı insana ait olup olmadığı söylenebilecek hale gelmektedir.

3. MALZEME VE YÖNTEM

Bu çalışmada kullanılan görüntüler içinde gerçekte yüz olan görüntülerdir. Gerçek dünya problemlerinde içinde yüz olmayan görüntülerin de veri tabanında bulunması ihtimali göz önünde bulundurulmalıdır. Fotoğrafta bulunan yüzlerin hangi oranda bulunabildiği başarımının yanında yüz olmayan görüntülerin kaçında yanlışlıkla yüz bulunduğu da gerçek dünya problemlerinin çözümü açısından önemlidir. Çalışmada kullanılan yüz bulan algoritmanın içinde yüz tespit ettiği fotoğraflar Şekil 3.1'de gösterilmiştir.





Şekil 3.1 Yüz tespit edilen fotoğraflar.

Şekil 3.1'de belirtilen şekillerde de görülebileceği gibi fotoğraflarda insan yüzü bulunmamasına rağmen yüz tespiti yapan algoritmanın yüz olarak algıladığı sınırlayıcı kutucuk içindeki pikseller algılanıp yüz tanıma sistemlerinde yüz arama listesine kaydedilebilmektedir. Arama uzayının kullanım amacı dışında gerçekte yüz olmayan piksellerden üretilmiş vektörlerle doldurulması sistemde performans kaybına yol açabilmektedir.

Bu çalışmada kullanılan görüntüler IJB-B veri kümesinin açık küme 1:N fotoğraflarda yüz kimliklendirme protokolünün gerçekleştirilmesi amacıyla IJB-B protokolünde verilen bilgilere göre kullanılmıştır. IJB-B veri kümesi kapsamında S1 galerisinde 3081 fotoğraf, S2 galerisinde 3376 fotoğraf olmak üzere aranacak kimlikler olarak 15199 fotoğraf kullanılmıştır. IJB-B protokolü şablon olarak adlandırılan bir kimliğe ait birden çok fotoğrafın kümesi anlamına gelen bir kavram kullanmaktadır. Şablonlar hem S1 ve S2 galerisinde hem de probe adı verilen aranacak fotoğraf listesinde kullanılmaktadır. CMC skoru kapalı küme kimliklendirme senaryosunun değerlendirilmesi için kullanılmaktadır. Aranacak fotoğraf listesindeki her bir şablon için o şablonun bilinen kimliğinin galeriden tahmin edilen listede ilk k sonuçta doğru pozitif görülüp görülmediğine göre hesaplanmaktadır. IET skoru ise yanlış pozitif kimliklendirme oranının (FPIR), yanlış negatif kimliklendirme oranına (FNIR) oranı olarak hesaplanmaktadır. FPIR, t düzeyinin üstünde probe aramasının tahmin listesi sonucunda gelen yanlış olan sonuçların skorudur. FNIR ise t düzeyinin üstünde probe aramasının tahmin listesinde gelmeyen ama gerçekte var olan eşleşmelerin skorudur. CMC kapalı küme 1:N performansını ölçerken IET açık küme 1:N performansını ölçmektedir. Probe arama süreleri ve şablon yaratımının ortalama süreleri de ek metrik olarak gösterilecektir.

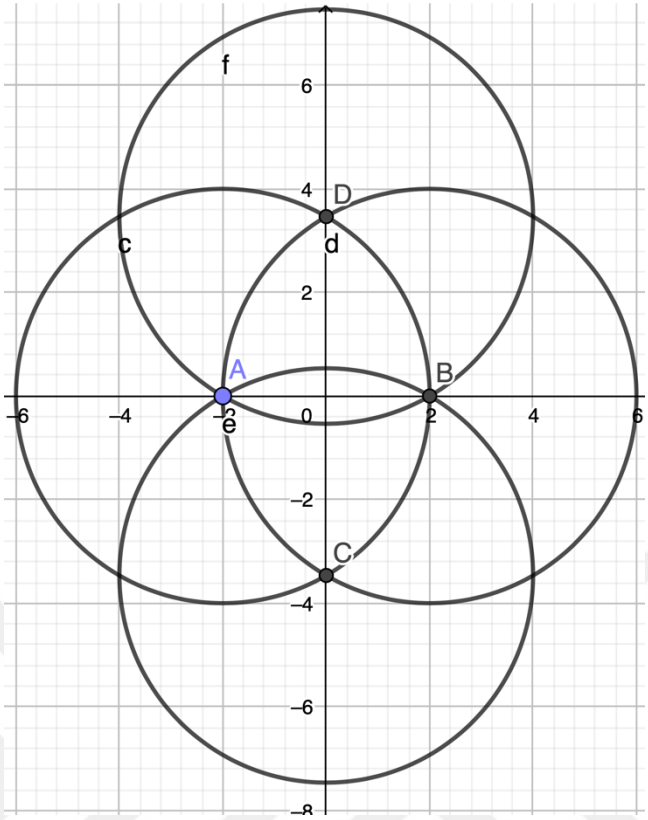
IJB-B veri kümesinde toplam 5732 fotoğraf kullanılarak oluşturulan 8104 şablon mevcuttur. 1845 farklı kimlik kullanılmıştır. S1 galerisinde 931 şablon ve kimlik, S2 galerisinde 814 kimlik kullanılmıştır.

Testin gerçekleştirilebilmesi amacıyla doğrudan SQL ve NOSQL veri tabanlarının destekleyebileceği bir veri modeli kullanılmıştır. Bu veri modellerinden ilki veri tabanına S1 ve S2 galerilerinden oluşturulan yüz vektörlerinin iki farklı tabloda tutulmasıdır. Liste veri yapısının veri tabanında uygulanışı bu şekildedir. Karşılaştırma yapılacak diğer veri tabanı modeli ise bir kök listeden oluşan derinliği iki olan bir ağaç yapısıdır. Veri aldıkça yatay genişleyen bu ağaç modelinde derinlik sayısı, kök listeye ekleme algoritmasıyla derinde bulunan listeye ekleme yaparken de devam ettirilerek artırılabilir. Veri yapısının derinliğinin sayısı kontrol edilmeyip 2 ile sınırlandırıldığında kök liste eleman sayısının 7600 olduğu durumda derin listedeki eleman sayısının 1560000 olduğu görülmüştür. Sınırlandırılma yapılmamış olsaydı tüm fotoğrafların kök listedeki bir elemanın altında benzer şartlarda dağılacağı kabul edildiğinde 200 derinliğine sahip bir ağaç elde edilebilirdi. 200 derinliğe sahip bir ağaçta veri yapısının sağlayacağı hızlandırma ve daraltacağı tahmin listesi arasında bir tercih yapılması gerekmektedir. Arama uzayını daha da daraltacak bu işlem aranan yüzle eşleşebilecek yüzlerle asla karşılaştırma yapılamadığı için CMC skorunu düşürme ihtimali vardır. Sadece 1000 yüz içinden ortalama 4 yüz listeyi genişlettiğinden ve kalan 996 yüz ağacı derinleştirmeye meyilli olduğundan ötürü 60 kata varan derinleşme veri yapısının oluşturulma algoritması sınırlandırılmadığı zaman daha avantajlı olabilmektedir.

Kullanılan yeni yöntem bir listede bulunan D boyuta sahip veri noktalarını aşağıda belirtilen şekilde bir veri yapısına dönüştürmektedir.

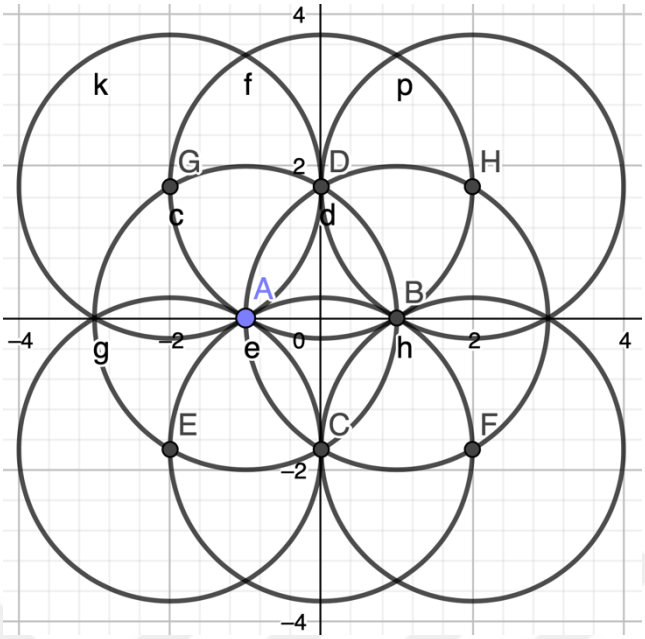
Belirtilen veri yapısında katmanlandırma kullanılmaktadır. Verilerin bulunduğu uzayda eş hacimleri kapsayacak veri noktalarıyla uzayın bölünmesi arama sırasında lineer uzayda yakın komşuların bulunmadığı bölgede birden çok veriyle karşılaştırmaktansa sadece tek veriyle karşılaştırma yapılıp o bölgedeki diğer tüm verilerle karşılaştırmaktan vazgeçilmektedir. Modelin avantajları ve dezavantajları aşağıda belirtilmiştir.

Çok boyutlu öklit uzayında noktaları dairesel-hacimsel şeklin merkezi olarak düşünürsek algoritma eş dairesel-hacimsel şekiller ile gelen veri uzayını tek bir katmanda sınırlı sayıda parçalara bölmektedir. Bir öklit uzayında herhangi bir noktadan boyut sayısının maksimum 2 katında eş dairesel-hacimsel şeklin çevresi temas edebilir (Şekil 3.2).



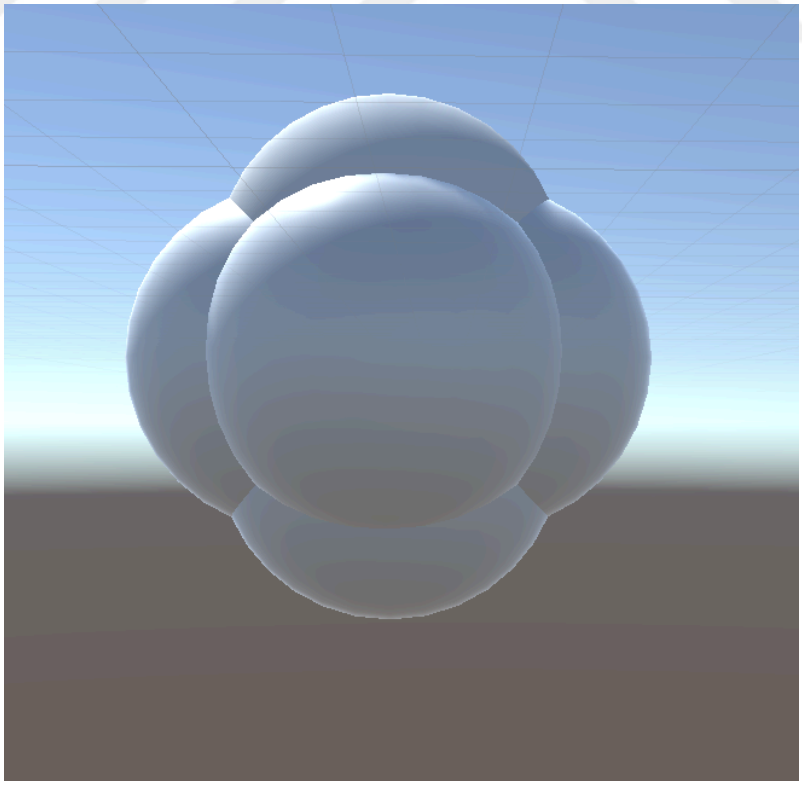
Şekil 3.2 2 boyutlu uzayda orjini kapsayabilen dairesel-hacimsel şekiller.

Şekil 3.2'deki gibi bir uzayda A, B, C ve D noktaları olduğu varsayıldığında (0,0) noktasında gelen bir arama A, B, C ve D noktalarıyla eşleşmektedir. Aynı uzayda G, H, F, E noktalarıyla olan öklit uzaklığı dairesel-hacimsel şeklin çapından büyük olduğundan dolayı eşleşme olmamaktadır. Dairesel-hacimsel şeklin yarı çap boyutuyla oluşturulma şekliyle öklit uzaklıklı eşleştirme eşiği eşit olduğu için yarı çapın boyutu sonucu etkilememektedir. Şekil 3.3'de orijin noktasının eşleştiği sadece 4 noktanın bulunduğu bir uzay gösterilmektedir.



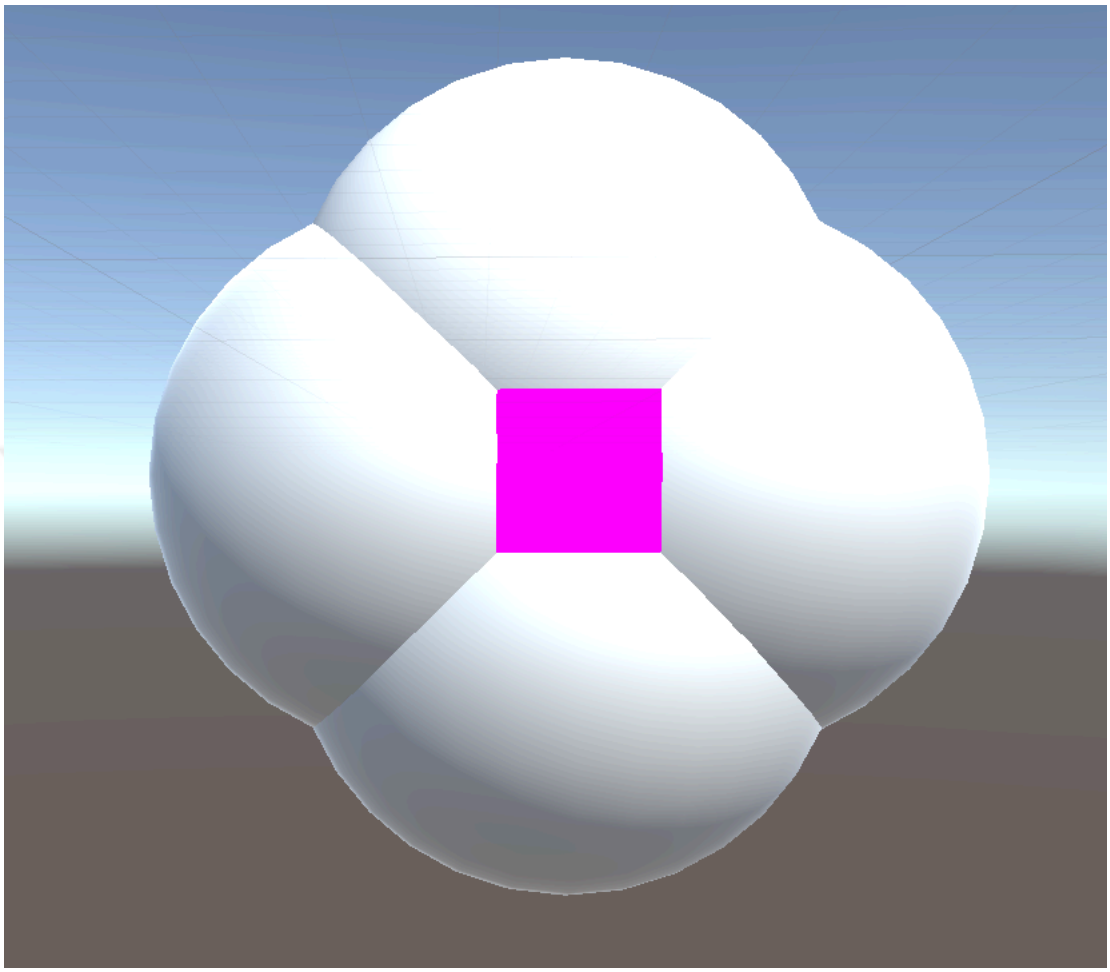
Şekil 3.3 2 boyutlu uzayda orjine yetişmeyen 4 nokta.

3 boyutlu uzay ele alındığında da boyut sayısının 2 katı kadar dairesel-hacimsel şeklin orijini kapsayabileceği görülür (Şekil 3.4).



Şekil 3.4 3 boyutlu uzayda orjini kapsayan 6 dairesel-hacimsel şekil.

Şekil 3.5'te gösterildiği gibi 3 boyutlu uzayda orijinde aynı yarıçapa sahip dairesel-hacimsel şekil eflatun renkle boyanmıştır.



Şekil 3.5 3 boyutlu uzayda orijinde aynı yarıçapa sahip dairesel-hacimsel şekil.

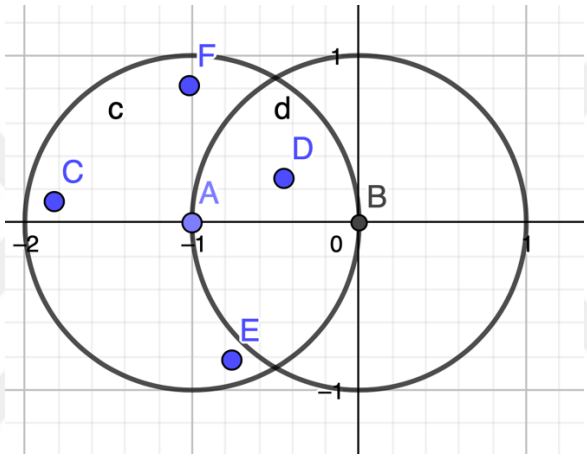
Şekil 3.5'de görülebileceği gibi orijinde Eflatun renkle boyalı alan iki boyutta taşmaktadır +z ve -z boyutunda oluşan bu taşan iki hacimsel bölgeye ancak iki tane daha dairesel-hacimsel şekil yerleştirilebilir.

Tüm liste elemanlarının katmanlandırılması durumunda 128 boyutlu verilerin bulunduğu veri yapısında tek katmanda bir arama maksimum 256 noktayla eşleşebilmektedir. Pratikte oldukça yoğun ve sık verilerin olduğu bir uzayda ancak görülebilecek bir senaryodur.

Sadece tek bir katman olsa dahi 7600 verinin olduğu kök listede en yüksek vakit alacak arama maksimum 256 noktayla eşleşerek altında yer alan verilerle arama sürdürülür. Bu da arama uzayını ortalama %97 daraltmaktadır. Bu daralma aranan verinin pozitif doğru olan yakın komşularının hiç aranmamasına da neden olmaktadır. Lineer liste aramasından daha hızlı

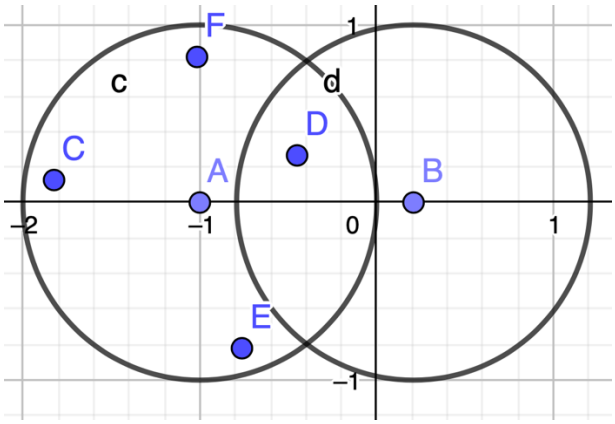
çalıştığı için yüksek oranlı eşleşmeler tekrar arama için kullanılabilmektedir. Liste aramada avantajı olmayan bu arama yöntemi aranan uzayın genişlemesine sebep olduğundan başarımı artırmaktadır.

Sadece A verisinin ana katmanda olduğu bir ağaç yapısında C, D, F ve E verileri A verisinin alt listesinde bulunsun. B noktasıyla bir arama yapıldığında B noktasının A'ya öklit uzaklığı yeterli yarı çapta olduğu için algoritma C, D, F ve E noktalarını B noktasıyla öklit uzaklığı cinsinden karşılaştırmaktadır. Eşleşen A ve D noktaları yakın komşuları bu senaryoda bulunabilmiştir (Şekil 3.6).



Şekil 3.6 B'nin yakın komşusu A ve D'nin bulunabildiği senaryo.

B noktası A noktasıyla eşleşebilecek uzaklıktan daha uzakta olursa fakat A noktasının altında yer alan bir D noktasıyla doğru pozitif olarak eşleşebileceği bir senaryoda ise arama algoritması D sonucunu tahmin listesine dahi alamamaktadır (Şekil 3.7).



Şekil 3.7 B'nin yakın komşusu D'nin üst katmandaki A noktasıyla eşleşmediği için bulunamadığı senaryo.

Bu sorunu aşmak amacıyla katmanlarda arama yaparken belirlenen sabit yarı çapın daha üstünde uzaklıkları kabul eden bir arama algoritması tasarlanabilmektedir. Fakat eşleşme sayısını ne kadar artıracağı belirsiz olduğundan bu arama uzaklığını belirlemek için ayrı bir çalışma yapılabilir.

Şekil 3.7'dan da görülebileceği gibi B ile eşleşmeyen F, C ve E noktaları doğrudan tahmin listesinden elenmiş haldedir.

Katmanlandırma sayısının sınırlandırılması işlemi de verilerin toplam veya beklenen boyutuna göre ayarlanabilmektedir. Katmanlandırma arttıkça arama daha dar bir uzayda gerçekleştirildiğinden doğru pozitif sayısı azalabilmektedir. Kullanılan CMC skoru rank-1 ve rank-10 skorları eşiğin üstünde yer alan doğru pozitif sayısıyla skorlama yaptığından ve milyonlarca verinin olduğu kümelerde aranan kimliğe ait fotoğraf sayısının da sınırlı olduğu göz önünde bulundurularak, katmanlandırmanın her katmanda arama uzayını ortalama %97 küçültmesi avantajlıdır. Eşiğin üstünde gelmeyen eşleşmesi gereken fotoğrafların artacak olması FNIR skorunu yükseltecektir. Eşik üstünde yanlış gelen fotoğraf sayısını da azaltacağından FPIR skoru düşecektir. FPIR'ın düşme oranının FNIR'ın yükselişinden daha yüksek olacağı çalışma öncesinde öngörülmüştür.

Ses veri tabanlarında benzer arama algoritmalarının lineer aramaya göre 109 kat hızlandırma, bölgesel hassas hash yöntemine göre 48 kat fazla olduğu çalışmalar mevcuttur. Fakat belirtilen yöntem veri tabanında herhangi bir sonuçla karşılaşmayan aramalarda etkili olmaktadır. [21]

Yaklaşık komşu bulmada kullanılan yöntemlerden Wu (2011) yüzlerin bölgelere ayrıldığı ve bölgelere kelime atanan bir indeksleme yöntemi kullanmıştır. Yan (2011) yüz özelliklerini ayırıcı uzaya hash fonksiyonu ile haritalayan bir regresyon algoritması kullanmıştır. Jegou (2010) uzayı düşük boyutlu alt uzaylara ayırıp bu alt uzayların Kartezyen çarpımını kullanmıştır [6][22][23].

Ponomarenko (2014) yaklaşık komşu bulma yöntemlerini karşılaştırmıştır. Yaklaşık komşu bulma yöntemleri verilerin sabit olduğu kabulünden yola çıkarak oluşturulmaktadır. Özelleştirilmedikçe *n* tane komşu bulma şeklinde bir sonuç döndürmektedirler. Belirli öklit uzaklığının altındaki tüm komşuları getirme problemi ise sonuçları değiştirmektedir [24].

Bölgesel hassas hash yöntemleri çok boyutlu verilerde verileri daha düşük verilerde kümelemek için gruplayıcı hash yöntemleri kullanmaktadır. Rasgele projeksiyon yöntemi bunlardan biri

olmakla birlikte genellikle listedeki veriler rasgele bir hiperdüzlem ile ikiye bölünmekte ve bu bölünmeler sonucunda bir ikili arama ağacı kullanmaktadır. Bu yöntem sabit sayıda verinin gruplanması için oldukça iyi sonuçlar verebilmekteyken verilerin sonradan arttığı veya rasgele farklı verilerin dahil olması gerektiği durumlarda ağacın baştan oluşturulması gerekmektedir. Hiperküre bazlı bölgesel hassas hash yöntemleri de bulunmaktadır. Terasawa (2017) hiperküre bölgesel hassas hash yöntemi önermiştir fakat önerisi verilerin birim hiperküre yüzeyinde olduğu verilere odaklanmaktadır. Yüz tanıma verileri ise birim hiperkürenin içinde yer almaktadır [25].

Bölgesel hassas hash yöntemlerinden optimum çözüme en yakın olanı uzayı hiperkürelere bölmektedir [26]. Bu çalışma kapsamında kullanılan yöntemde uzayı hiperkürelere bölmek yerine uzayda var olan veri noktalarının kendisi uzayı bölücü hiperküreler olarak kullanmaktadır. Bunun avantajlarından bir tanesi uzayı bölmek için gereken hiperküre sayısının doğal olarak sınırlandırılmasıdır. Model sadece verilerin kendisini hafızada tuttuğundan verilerin kendisi dışında ekstra tutulması gereken tek şey katmanlar arası geçiş işaretçileridir. Bir çok veriyi kapsayan sanal hiperküreler de kurulabilir fakat bu işlem gelecek verileri kapsayacak optimum sanal hiperküreyi bilmek imkansız olduğu için kullanılması zordur. Bölgesel hassas hash yöntemleri çoklu veride işlem yapmak için uygundur. Tekli veride yapılacak bir atomik işlem sonucu çoklu aramaya imkân veren bir algoritma değildir. Sanal hiperküreleri kapsadığı verilere bakarak onları optimum derecede temsil edecek şekilde güncellemek de mümkündür. Fakat böyle bir durumda ana katmandaki bölgeleri temsil eden sanal hiperküreler bir diğerinin alanını işgal etmeye başlar ve bu kuralın ihlali verilerin uzayının eş parçalara bölünmek yerine katmanlandırmadaki kümeleri temsil eden sanal hiperkürelerin arama sürelerini artırmasına yol açmaktadır.

Çalışmadaki veri yapısı algoritmasında bulunmayan fakat eklenebilecek bir diğer özellik yüz verisini katmanlandırma yaparken sadece eşleştiği en yüksek katman verisinin altındaki listeye eklemek değil eşleştiği her verinin altındaki listeye eklemektir. Böylesi bir durumda *d* boyuta sahip bir veri maksimum *2*d* sayısı kadar farklı listede bulunabilecektir. Böylesi bir durumda veriyi sadece tek bir yerde tutup veriye ait göstergeçler listelere eklenebilir. Verinin bulunabilme ihtimalini arttıracak olsa da belirtilen çözüm yakın komşuları %100 kesinlikle getirmemektedir.

Çalışma kapsamında, IJB-B veri kümesinin açık küme 1:N fotoğraflarda yüz kimliklendirme protokolünün testi iki farklı algoritmayla gerçekleştirilerek sonuçlar karşılaştırılmıştır. Bu test

ikisi birbirinden farklı arama galerisi ve aranacak kimlikleri içeren bir küme olarak üç farklı küme içermektedir. Arama galerileri S1 ve S2 birbirinden bağımsız kimliklere ait fotoğraflar içeren iki farklı arama listesidir. Probe ismi verilen üçüncü kümedeki her bir fotoğraftaki yüzler tespit edilip S1 ve S2 galerilerinde arama yapılmıştır. Arama sonucunda oluşan tahmin listesinde Probe kümesinde aranan kimliğe ait fotoğrafların listede hangi sıralarda geldiği değerlendirilerek CMC, FPIR, FNIR, Rank-1 ve Rank-10 ve yeni önerilen Rank-N Total (R-NT) puanları karşılaştırılmıştır [9]. Puanları karşılaştıran iki farklı algoritmadan biri lineer benzerlik araması yaparken çalışma kapsamında önerilen yöntem yaklaşık yakın komşu araması yapmaktadır.

İki algoritmanın da bu farklılık dışında ortak kullandığı yöntemler ve izlediği işlemler aşağıda gösterilmiştir:

A) Lineer Öklid araması algoritması için izlenen işlem adımları:

A.1.) S1 ve S2 galerilerindeki fotoğraflardaki her bir yüz HOG özellikleri kullanılarak bulunup 29 katmanlı ResNet derin öğrenme modeli kullanılarak 128 boyutlu vektöre dönüştürülmektedir. ResNet derin öğrenme modeli ResNet-34 ağındaki bazı katmanların kaldırıp, katmandaki filtrelerin yarıya düşürülmesiyle elde edilmiştir ve 3 milyon yüzle öğretilmiştir. 3 milyon yüz Face Scrub[18] ve VGG[19] veri kümeleri kullanılarak elde edilmiştir. Veri kümesinde 7485 farklı kimlik bulunmaktadır. Bu model LFW[20] değerlendirme protokolünde 1:1 karşılaştırmada %99.38 başarım elde etmiştir.

A.2.) S1 ve S2 galerilerinden elde edilen her bir 128 boyutlu vektör ait oldukları kimlik bilgileriyle birlikte her iki yöntemde de ayrı iki tabloda düz bir listenin elemanları şeklinde kaydedilmiştir.

A.3.) Probe kümesindeki her bir fotoğraftaki yüz HOG özellikleri kullanılarak bulunup 29 katmanlı ResNet derin öğrenme modeli kullanılarak 128 boyutlu vektöre dönüştürülmektedir ve kimlik bilgileriyle birlikte arama tablosuna kaydedilmiştir.

A.4.) Arama tablosundaki her bir verideki 128 boyutlu vektör sırasıyla S1 ve S2 galerilerinden elde edilen iki farklı tablodaki verilerin 128 boyutlu vektörleriyle teker teker öklit hesabıyla karşılaştırılıp 0.6'nın altı sonuç verenler iki farklı tahmin listesinde küçükten büyüğe sıralı şekilde saklanmıştır.

A.5.) İki farklı tahmin listesindeki her bir arama verisinde S1 ve S2 galerilerinde olduğu bilinen kimliklerin fotoğraflarının kaç tanesinin geldiği ve hangi sırada geldiği tespit edilmiştir.

A.6.) Bu bilgiler değerlendirilerek CMC, FPIR, FNIR, Rank-1 ve Rank-10 ve yeni önerilen Rank-N Total (R-NT) puanları tespit edilmiştir [9].

B) Yaklaşık yakın komşu araması algoritması için izlenen işlem adımları:

B.1.) A.1. ile aynı işlem yapılmıştır.

B.2.) S1 ve S2 galerilerinden elde edilen her bir 128 boyutlu vektör ait oldukları kimlik bilgileriyle birlikte şu şekilde kaydedilmiştir:

B.2.1.) Her bir galeri için:

B.2.2.) Galeride henüz eklenmemiş her bir 128 boyutlu vektör için:

B.2.3.) O galeriye ait ağaç veri yapısında kök listedeki her bir elemanla öklit uzaklığıyla karşılaştır, kök listede hiç eleman yoksa kök listeye eleman olarak ekle.

B.2.4.) Kök listede eleman varsa ağaç veri yapısında kök listede öklit uzaklığı sonucu 0.6 altında bulunarak eşleşen ilk elemanın çocuğu olan listede sona ekle.

B.2.5.) Kök listede eleman varsa ve ağaç veri yapısında kök listede öklit uzaklığı sonucu 0.6 altında eşleşen hiç eleman bulunmuyorsa kök listede sona ekle.

B.3.) Probe kümesindeki her bir fotoğraftaki yüz HOG özellikleri kullanılarak bulunup 29 katmanlı ResNet derin öğrenme modeli kullanılarak 128 boyutlu vektöre dönüştürülmektedir ve kimlik bilgileriyle birlikte arama tablosuna kaydedilmiştir.

B.4.) Arama tablosundaki her bir verideki 128 boyutlu vektör sırasıyla S1 ve S2 galerilerinden elde edilen iki farklı ağaç veri yapısındaki verilerin 128 boyutlu vektörleriyle şu şekilde öklit uzaklığı hesaplanarak tahmin listesi oluşturulmuştur:

B.4.1.) Arama tablosundaki her bir veri için:

B.4.2.) Sırasıyla S1 ve S2 galerilerine ait ağaçlarda arama yap.

B.4.3.) Ağaçtaki kök listede her bir liste elemanıyla öklit uzaklığı hesabı yap, uzaklığı 0.6 altı çıkan her bir elemanı sıralı tahmin listesine ekle. Eklenen her bir elemanın çocuğu olan her bir listedeki elemanla öklit uzaklığı hesabı yap uzaklığı 0.6 çıkan her bir elemanı sıralı tahmin listesine ekle.

B.5.) İki farklı tahmin listesindeki her bir arama verisinde S1 ve S2 galerilerinde olduğu bilinen kimliklerin fotoğraflarının kaçının geldiği ve hangi sırada geldiği tespit edilmiştir.

B.6.) Bu bilgiler değerlendirilerek CMC, FPIR, FNIR, Rank-1 ve Rank-10 ve yeni önerilen Rank-N Total (R-NT) puanları tespit edilmiştir [9].

4. BULGULAR

Çalışma kapsamında IJB-B veri kümesi üzerinde lineer arama yöntemi ve yeni yöntem kullanılarak performans değerleri ISO/IEC 19795-1 standartlarında belirtildiği şekilde raporlanmıştır [27].

Bu bölümde kullanılan yöntemlerin sonuçlardan bahsedilerek yöntemler Tablo 4.1'de karşılaştırılmıştır.

Tablo 4.1 Kimlik tanımlamada galeri ve arama bilgisine göre problemlerin gruplandırılması.

		Aranacak Kimliğe ait bir fotoğraf mı var?	
		Bir fotoğraf	Çok Fotoğraf
Galerideki fotoğrafların kime ait olduğu biliniyor mu? Bir fotoğraf mı var?	Kimlikleri bilinen ve bir kimliğe ait tek fotoğrafın olduğu çok kimlikli galeri.	Galeride en yakın temsili bulma problemi.	Çok fotoğrafla istatistiki olarak en çok eşleşen temsili bulma problemi.
	Kimlikleri bilinen ve bir kimliğe ait çok fotoğrafın olduğu çok kimlikli galeri.	Sınıflandırma problemi, destek vektör makinesi kullanılabilir.	Sınıflandırma problemi çözüldükten sonra çok fotoğrafla en çok eşleşen sınıfları bulma problemi.
	Kimlikleri bilinmeyen ve bir kimliğe ait tek fotoğrafın olduğu galeri.	Galeride en yakın temsili bulma problemi.	Çok fotoğrafla istatistiki olarak en çok eşleşen en yakın temsili bulma problemi.
	Kimlikleri bilinmeyen ve bir kimliğe ait çok fotoğrafın olduğu galeri.	Galeriden bulunacak en yakın temsillerin, kimliği aranan şahsa ait olması için doğru sıralama önemlidir. İdeal olan tüm temsilleri yüksek sırayla getirmektir.	Galeriden bulunacak en yakın temsillerin, kimliği aranan şahsa ait olması için doğru sıralama önemlidir. Fotoğraflarla en çok eşleşenleri bulabilmek gerekir.

Tablo 4.1'de de görülebileceği gibi üzerinde önceden kaydedilmiş kimlikleri bilinen birden çok fotoğraf olması senaryosu başarımın en yüksek olduğu senaryodur. Hem gerçeğe ait çok veri bulunması hem de problemin benzerlik aramasından sınıflandırma aramasına dönüşmesi bunların nedenlerinden biridir. Bu unsurların yanında Tablo 4.2'de aranacak kimliğin galeride olduğu bilgisinin bilinmeyişinin probleme etkisi gösterilmiştir.

Tablo 4.2 Açık-Küme, Kapalı-Küme problemi.

		Aranacak Kimliğin Galeride Olduğu kesin mi?	
		Evet – Kapalı Küme	Hayır – Açık Küme
Galerideki fotoğrafların kime ait olduğu biliniyor mu? Bir fotoğraf mı var?	Kimlikleri bilinen ve bir kimliğe ait tek fotoğrafın olduğu çok kimlikli galeri.	Galeride en yakın temsili bulup, zaten var olan gerçek kimlikle karşılaştırmak problemi doğrudan çözmektedir.	Kimlikler bilindiği için kimliklere ait bilgiler işlenip, bu galeriye ait bir kimlik yok denebilmelidir.
	Kimlikleri bilinen ve bir kimliğe ait çok fotoğrafın olduğu çok kimlikli galeri.	Sınıflandırma probleminde en yakın sınıfı bulmak, en yakın temsili bulmaktan daha başarılı olmaktadır.	Kimlikler bilindiği için kimliklere ait bilgiler işlenip, bu galeriye ait bir kimlik yok denebilmelidir.
	Kimlikleri bilinmeyen ve bir kimliğe ait tek fotoğrafın olduğu galeri.	Galeride en yakın temsili bulmak yeterli olmayabilir. Benzerlik listesi gerektirir.	Kimlik galeride yok veya benzerleri şunlardır denebilmelidir.
	Kimlikleri bilinmeyen ve bir kimliğe ait çok fotoğrafın olduğu galeri.	Benzerlik listesi gerektirir.	Kimlik galeride yok veya benzerleri şunlardır denebilmelidir.

Yüz tanıma probleminin belirtilen boyutlarda farklı sonuçlar gerektirmesi alanı karmaşıklaştırmıştır. Yüz tanıma çözümleri çoğu zaman bire bir kimlik doğrulama işlemiyle anılmaktadır. %90'ın üzerine çıkan başarım gösteren algoritmaların hepsi bire bir iki fotoğrafın aynı kimliğe ait olup olmadığı problemini çözmektedir.

Çalışma kapsamında kullanılan IJB-B 'nin açık küme 1:N durağan fotoğraf protokolü kimlikleri bilinmeyen ve bir kimliğe ait çok fotoğrafın olduğu galeride, aranacak kimliğe ait çok fotoğrafın bulunduğu, aranacak kimliğin galeride olup olmadığının bilinmediği senaryoyu işlemektedir. Problemin çözümünün gereksinimi şu şekildedir:
1) Galeriden bulunacak en yakın temsillerin kimliği aranan şahsa ait olması için doğru sıralama önemlidir. İdeal olan tüm temsilleri yüksek sırayla getirmektir.

2) Kimlik galeride yoksa tahmin listesinin boş dönmesi beklenmektedir.

3) Aranacak kimliğe ait çok fotoğrafın olması tek fotoğraf olmasından daha iyi bir senaryodur. İstatistiki metodlarla daha başarılı sonuçlar elde edilebilir. Aranacak kimliğe ait tek fotoğraf olsaydı kimliğe ait bilgi daha kısıtlı olacağı için yapılabilecekler de sınırlı olmaktadır.

IJB-B'de verilen galerilerde fotoğrafın kime ait olduğu bilgisi bulunmasına rağmen kullanılmamıştır. IJB-B ve IJB-C sınıflandırma ve kümeleme için ayrı bir protokol sunduğundan dolayı kullanılan protokolde verilen kimlik bilgisi test sonuçlarını doğrulamak için kullanılmıştır. IJB-B veri kümesinde aranacak kimlik fotoğrafları listesi şu şekilde oluşturulmuştur:

1) Galeride olmayan tüm fotoğraflardaki her bir kimlik tek fotoğrafa sahip arama kimliği olarak eklenmiştir.
2) Galeride olmayan tüm fotoğraflar yarıya bölünmüş ve iki yarı da birden çok fotoğraf içeren arama kimliği olarak eklenmiştir.
3) Galeride olmayan tüm fotoğraflar birden çok fotoğraf içeren arama kimliği olarak eklenmiştir.

Arama kimliği havuzu Mongo NOSQL veri tabanında tutulmuştur. Çift çekirdekli Intel Broadwell E5-2686v4 işlemciye sahip 8 GB bellekli düşük-orta sınıf ağ performansına sahip bir bilgisayar aramaları bu veri tabanından veri çekerek kendi işlemcisinde gerçekleştirmektedir. Daha az veri çekilmesinin yanında daha az işlem ve daha az karşılaştırma yapıldığından, yeni yöntem daha hızlı çalışmaktadır.

Arama kimliği havuzunda bulunan toplamda 15199 tane yüz ve 8104 şablon iki galeride aratılmış ve sonuçlar aşağıdaki gibi bulunmuştur.

Hız:

16 şablonla yapılan aramada tüm listenin lineer aranması 48.659 saniye, yeni yöntem ile aranması 19.940 saniye sürmüştür. Bu aramanın doğru pozitif fotoğrafın bulunduğu sıraları içeren sonuçları Tablo 4.3'te gösterilmiştir.

Tablo 4.3 Tek Senaryo için örnek sonuçlar.

Sonuçlar:	Lineer arama için rank-1 rank-50 arası doğru pozitif fotoğrafın bulunduğu sıraya göre veriler:							
Lineer	2	3	3	5	16	17	17	23
Yeni Yöntem	3	5	6	6	9	22	34	50

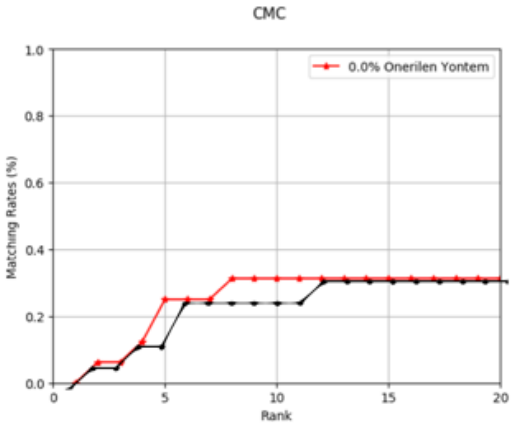
İlk 50'de yeni yöntemin rank-50'ye göre R-NT puanı 10,75'tir.

İlk 50'de listede lineer taramanın rank-50'ye göre R-NT puanı 18,125'tir.

İki yöntemin de ilk 50'de 8 doğru pozitif getirmesi aratılan şablonlara özgü bir durumdur.

Yeni öne sürülen Rank N Toplam R-NT metriği bir tahmin listesinin ilk N sırasında gelen doğru pozitiflerin sıra toplamlarının o aralıkta gelen doğru pozitiflere oranıdır.

Tablo 4.3'te yer alan veriler kullanılarak çizilen CMC grafiği Şekil 4.1'de gösterilmiştir:



Şekil 4.1 CMC Skoru karşılaştırması.

İki yöntemde de kalan 8 şablonun doğru cevapları ilk 50 sonuçta görülememiştir.

4.1. PERFORMANS DEĞERLENDİRMESİ

FRVT raporlamasında ve IJB-B veri kümesinde belirtilen CMC, FPIR, FNIR, TPIR, Rank-1, Rank-10, Rank-20, Rank-50, FNIR@FPIR=0,2 metriklerinin, IET grafiği hesaplanışı ve SELECTIVITY ile ilgili bilgilere bu bölümde değinilmektedir [7][8] [28].

- CMC, bir arama kümesinde aranan kimliklere ait fotoğrafın varlığı kesinse tüm aramalardan yüzde kaçının doğru pozitif sonuç verdiğini,
- FPIR, bir arama kümesinde aranan kimliklere ait fotoğrafın olmadığı kesin olarak bilinen aramalarda yanlış adaylar içeren tahmin listesinin arama sonuçlarının tüm aramaların yüzde kaçında olduğunu,
- SELECTIVITY, bir arama kümesinde aranan kimliklere ait fotoğrafın olmadığı kesin olarak bilinen aramalarda yanlış adaylar içeren tahmin listesinin toplam eleman sayısının arama sonuçlarının toplam aramalara oranını,
- FNIR, bir arama kümesinde aranan kimliklere ait fotoğrafın olduğu kesin olarak bilinen aramalarda tahmin listesinin aranan kimliğe ait doğru adayı içermediği arama sonuçlarının tüm aramaların yüzde kaçında olduğunu,
- Rank-1 ve Rank-N, N tahmin listesindeki sırayı tam sayı olarak belirtmek üzere tahmin listesinin belirtilen ilk N sırada aranan kimliğe ait doğru adayın bulunduğu arama sonuçlarının tüm aramaların yüzde kaçında olduğunu,
- FNIR@FPIR=0,2 FPIR değerinin 0,2 değerini aldığı sonuçlarda FNIR'ın aldığı değeri,
- FNIR(N,R,T,L) ve FPIR(N,T,L) fonksiyonlarındaki N değeri galerideki toplam fotoğraf sayısını L değeri tahmin listesini R değeri tahmin listesindeki sırayı ve T değeri tahminlerin sahip olduğu benzerlik oranını,
- IET grafiği FNIR@FPIR türünde metriklerin hesaplandığı algoritmaları karşılaştırmak için kullanılan grafiği tanımlamaktadır.

FPIR(N,T,L) = Eşleşmenin Kesin Olmayacağı Aramalarda Yanlış Adayların Doğru Kabul Edildiği Arama Sayısı / Eşleşmenin Kesin Olmayacağı Arama Sayısı

FNIR(N,R,T,L) = Eşleşmenin Olabileceği Aramalarda Doğru Adayların Yanlış Kabul Edildiği Arama Sayısı / Eşleşmenin Olabileceği Arama Sayısı

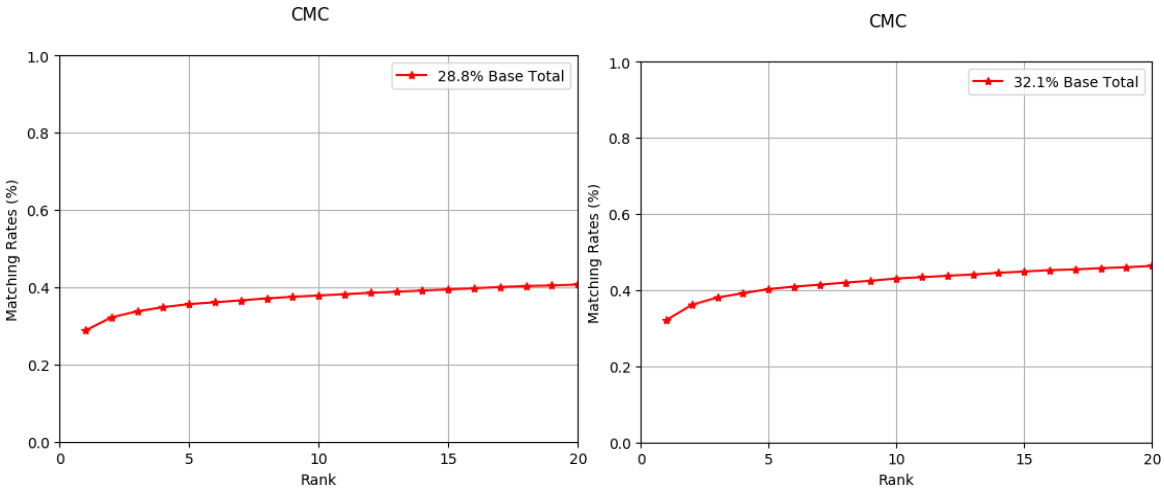
FPIR değeri sabit tutularak FNIR'daki değişimler gözlemlenmektedir. FPIR değerini sabit tutmanın yöntemi T benzerlik oranını uygun şekilde değiştirmektir. T benzerlik oranı

değiştiğinde FNIR değerini de değiştirmektedir. T benzerlik oranı kabul gereksinimi artırılırsa daha az aday kabul edilecektir böylece FNIR değeri artmaktadır. Kabul oranı gereksinimini artırmak FPIR değerini ise azaltmaktadır. Yüz tanıma algoritmalarının çoğunda bu değer değişimleri gözlemlenerek 1:N açık küme problemlerindeki performansı karşılaştırılmaktadır. Aynı FPIR değerlerinde daha düşük FNIR değerine sahip olan algoritma daha az sayıda aranacak kümede kimliği kesin bulunan ama tahmin listesinde getirilemeyen sonuç üretecektir. FPIR değeri sonuç getirmemesi gerekmesine rağmen yanlış adayları tahmin listesinde getiren algoritmanın açacağı sorunu temsil etmektedir. Örneğin pasaport geçişlerinde suçluların bulunduğu listeyle eşleşen bir yanlış şüpheli ne kadar çok suçluyla yanlış pozitif eşleşirse orada çalışan polisin daha fazla sayıda eşleşmeyi tek tek incelemesi gerekmektedir. O sebeple FNIR@FPIR=0,2 gibi grafiklerde daha iyi algoritma Y sabitken daha düşük X değerine, X sabitken daha düşük Y değerine sahip olan algoritmadır. SELECTIVITY ise FPIR değerinin özel bir hali olarak algoritmaların yanlış pozitif liste boylarının niceliklendirilmesi için kullanılmaktadır. FPIR değeri polisin kaç farklı kişide aslında hiç olmayan bir kimliğin eşleşmelerini aradığını niceliklendirirken, SELECTIVITY aslında hiç arama sonucunda olmayan bir kimliğin kaç eşleşmesini aradığını niceliklendirmektedir.

T değeri sonsuz olsa dahi sistemde varlığı bilinen yüzler yüz tespit eden algoritma tarafından yüz kabul edilmeyebildiği için sistemde yüzün fotoğrafta bulunması probleminde de belirli bir hata yüzdesi bulunmaktadır.

33 kimlikle yapılan bir aramada T değeri sonsuz olmasına rağmen 4 kimliğe ait veriler veri tabanında bu sebepten dolayı bulunamamıştır.

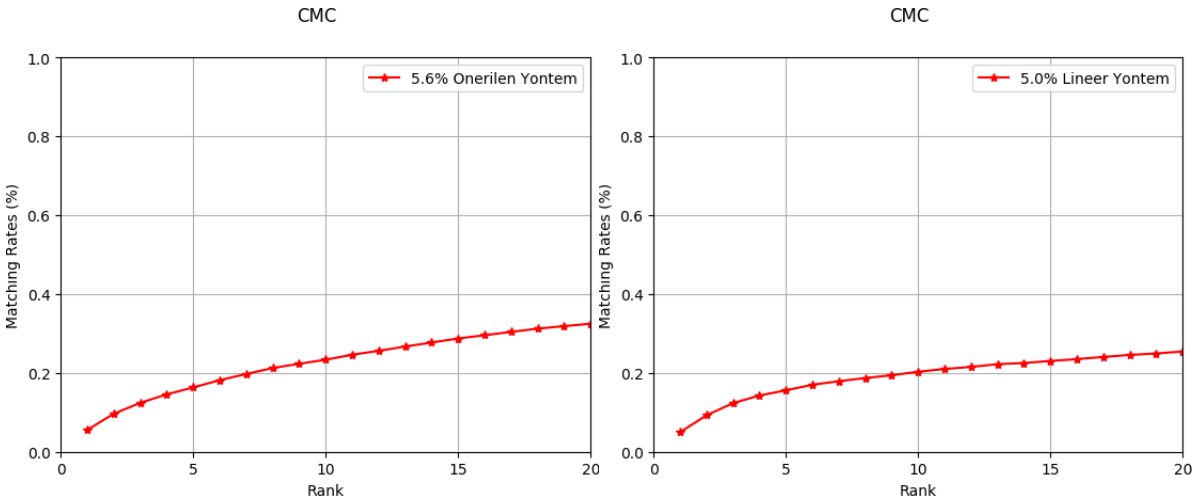
8103 kimlikle yapılan tüm veri kümesini içeren aramada elde edilen sonuçlar Şekil 4.2’de gösterilmiştir:



Şekil 4.2 Yeni yöntem(sol) CMC grafiği Lineer liste araması(sağ).

Her bir kimlik aramasının en az bir doğru pozitif sonuç verdiği sıra göz önünde bulunarak çizilen CMC grafiği Şekil 4.2.'te gösterilmiştir. Lineer liste araması aynı sıralamalarda daha fazla kimlik aramasında en az bir doğru pozitif sonuç vermesi açısından daha başarılıdır.

Her bir kimlik aramasının hiç yanlış negatif sonuç vermediği ilk sıralama göz önünde bulunarak çizilen CMC grafiği Şekil 4.3.'te gösterilmiştir:



Şekil 4.3 Yeni yöntem(sol) CMC grafiği Lineer liste araması(sağ) tüm doğru pozitifler.

Şekil 4.3'te gösterildiği gibi yeni yöntem aynı sıralamalarda lineer aramadan daha fazla kimlik aramasında tüm doğru pozitifleri kapsaması açısından daha başarılıdır.

CMC grafiklerinin çizilebilmesi için aynı T eşik değerinde Lineer arama ve yeni yöntemin iki farklı versiyonunun IJB-B veri kümesinde verdiği sonuçlar Tablo 4.4'de gösterilmiştir.

Tablo 4.4 T 0.60 değerinde tüm kümenin arama sonuçları.

T=0.60	Yeni Yöntem	Lineer Arama	Yeni Yöntem (Tekrar Aramasız)
R-1 Hit	0.2879 (Miss 0.712)	0.321 (Miss 0.678)	0.265 (Miss 0.712)
R-10	0.378	0.430	0.373
R-20	0.4066	0.463	0.4023
R-50	0.448	0.518	0.443
FNIR	0.485	0.381	0.491
FPIR	0.935	0.995	0.932
SELECTIVITY	88.127	132.005	79,46
FNIR-ALL	0.798	0.716	0.80
TPIR	0.5145	0.618	0.508
Önerilen FNIR	0.642	0.548	0.646
RN-T	0.7465	0.992	0.675
Çalışma Zamanı (dakika)	1042	1166	368

Tablo 4.4'te gösterildiği gibi standart lineer arama yöntemi kullanıldığında, sırasıyla %32.1, %43, %46.3, %51.8, %38.1, %99.5, 132, %71.6, %61.8, %54.8, %99.2 ve 1166 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR, RN-T ve Çalışma Zamanı değerleri tespit edilmiştir. Yeni yöntem kullanıldığında, sırasıyla %28.79, %37.8, %40.66, %44.8, %48.5, %93.5, 88.127, %79.8, %51.45, %64.2, %74.65 ve 1042 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR, RN-T ve Çalışma Zamanı değerleri tespit edilmiştir. Yeni yöntemin veri kümesinde yüksek ihtimalle doğru bulduğu düşünülen oranlarda eşleşen yüzlerle tekrar arama yapılmayan versiyonu kullanıldığında, sırasıyla %26.5, %37.3, %40.23, %44.3, %49.1, %93.2, 79.46, %80, %50.8, %64.6, %67.5 ve 368 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR,

Önerilen FNIR, RN-T ve Çalışma Zamanı değerleri tespit edilmiştir. Tüm sonuçlar Tablo 4.4'de gösterilmiştir.

IET grafiğinin çizilmesi için aynı T eşik değerli ölçümlerin dışında aynı FPIR değerlerine sahip farklı T değerlerinde yöntemlerde tespit edilecek FNIR değeri tespit edilmelidir. Grafiğin çizilip karşılaştırılabilmesi için en az üç ölçüm gerekmektedir. Bu üç ölçümün gerçekleştirilebilmesi amacıyla FPIR değerlerinin sırasıyla 0.2, 0.02 ve 0.002 olarak tespit edildiği farklı T eşik değerlerindeki ölçümler gerçekleştirilmiştir.

8103 kimlikle yapılan tüm veri kümesini içeren FPIR~=0.2 değerli aramada elde edilen sonuçlar Tablo 4.5'te gösterilmiştir:

Tablo 4.5 FPIR 0.2 değerinde tüm kümenin arama sonuçları.

FPIR~=0.2	Yeni Yöntem	Lineer Arama	Yeni Yöntem (Tekrar Aramasız)
R-1 Hit	0.199 (Miss 0.80)	0.209 (Miss 0.790)	0.191 (Miss 0.808)
R-10	0.223	0.235	0.229
R-20	0.225	0.236	0.232
R-50	0.228	0.236	0.232
FNIR	0.771	0.763	0.767
FPIR	0.20	0.20	0.20
SELECTIVITY	2,548	0,846	1,354
FNIR-ALL	0.939	0.941	0.923
TPIR	0.228	0.236	0.232
Önerilen FNIR	0.939	0.941	0.923
RN-T	0,134	0.064	0,166

Tablo 4.5'te gösterildiği gibi FPIR 0.2 değerini veren farklı T eşik değerlerinde Lineer arama ve yeni yöntemin iki farklı versiyonunun IJB-B veri kümesinde verdiği sonuçlar Tablo 4.5'de gösterilmiştir. Standart lineer arama yöntemi kullanıldığında, sırasıyla %20.9, %23.5, %23.6,

%23.6, %76.3, %20, 0.846, %94.1, %23.6, %94.1, %6.4 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntem kullanıldığında, sırasıyla %19.9, %22.3, %22.5, %22.8, %77.1, %20, 2,548, %93.9, %22.8, %93.9, %13.4 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntemin veri kümesinde yüksek ihtimalle doğru bulduğu düşünülen oranlarda eşleşen yüzlerle tekrar arama yapılmayan versiyonu kullanıldığında, sırasıyla %19.1, %22.9, %23.2, %23.2, %76.7, %20, 1.354, %92.3, %23.2, %92.3, %16.6 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Tüm sonuçlar Tablo 4.5'de gösterilmiştir. FPIR 0.02 değerini veren farklı T eşik değerlerinde Lineer arama ve yeni yöntemin iki farklı versiyonunun IJB-B veri kümesinde verdiği sonuçlar Tablo 4.6'da gösterilmiştir.

Tablo 4.6 FPIR 0.02 değerinde tüm kümenin arama sonuçları**.**

FPIR~=0.02	Yeni Yöntem	Lineer Arama	Yeni Yöntem (Tekrar Aramasız)
R-1 Hit	0.125 (Miss 0.875)	0.119 (Miss 0.88)	0.109 (Miss 0.890)
R-10	0.13747	0.12896	0.1250
R-20	0.13745	0.12896	0.1261
R-50	0.13745	0.12896	0.1264
FNIR	0.860	0.871	0.8735
FPIR	0.02	0.02	0.02
SELECTIVITY	0.80	0,053	0,096
FNIR-ALL	0.962	0.974	0.965
TPIR	0.139	0.1289	0.126
Önerilen FNIR	0.962	0.974	0.965
RN-T	0,0877	0,0184	0,070

Tablo 4.6'da gösterildiği üzere standart lineer arama yöntemi kullanıldığında, sırasıyla %11.9, %12.896, %12.896, %12.896, %87.1, %2, 0,053, %97.4, %12.89, %97.4, %1.84 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntem kullanıldığında, sırasıyla %12.5, %13.747, %13.745, %13.745, %86, %2, 0,80, %96.2, %13.9, %96.2, %8.77 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntemin veri kümesinde yüksek ihtimalle doğru bulduğu düşünülen oranlarda eşleşen yüzlerle tekrar arama yapılmayan versiyonu kullanıldığında, sırasıyla %10.9, %12.50, %12.61, %12.64, %87.35, %2, 0,096, %96.5, %12.6, %96.5, %7 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Tüm sonuçlar Tablo 4.6'da gösterilmiştir.

FPIR 0.002 değerini veren farklı T eşik değerlerinde Lineer arama ve yeni yöntemin iki farklı versiyonunun IJB-B veri kümesinde verdiği sonuçlar Tablo 4.7'de gösterilmiştir.

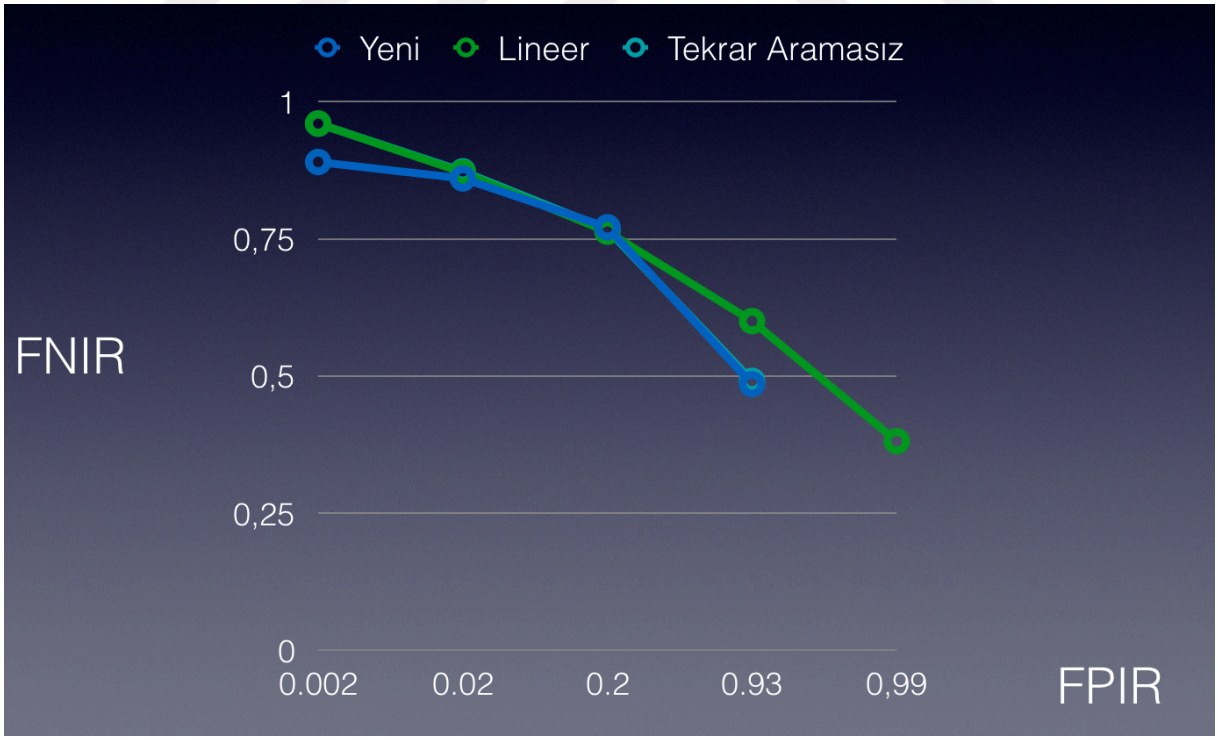
Tablo 4.7 FPIR 0.002 değerinde tüm kümenin arama sonuçları.

FPIR~=0.002	Yeni Yöntem	Lineer Arama	Yeni Yöntem (Tekrar Aramasız)
R-1 Hit	0.095 (Miss 0.904)	0.0362 (Miss 0.963)	0.0338 (Miss 0.904)
R-10	0.1061	0.03776	0.0378
R-20	0.1066	0.03776	0.0378
R-50	0.1066	0.03776	0.0378
FNIR	0.8933	0.962	0.962
FPIR	0.002	0.002	0.002
SELECTIVITY	0,389	0,0058	0.218
FNIR-ALL	0.969	0.993	0.988
TPIR	0.106	0.037	0.0378
Önerilen FNIR	0.969	0.993	0.988

RN-T	0,0781	0,0040551	0,0234

Tablo 4.7'de gösterildiği gibi FPIR 0.002 değerini veren farklı T eşik değerlerinde Lineer arama ve yeni yöntemin iki farklı versiyonunun IJB-B veri kümesinde verdiği sonuçlar Tablo 4.7'de gösterilmiştir. Standart lineer arama yöntemi kullanıldığında, sırasıyla %3.62, %3.776, %3.776, %3.776, %96.2, %0.2, 0,0058, %99.3, %3.7, %99.3, %0.40551 R-1, R-10, R-20, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntem kullanıldığında, sırasıyla %9.5, %10.61, %10.66, %10.66, %89.33, %0.2, 0,389, %96.9, %10.6, %96.9, %7.81 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Yeni yöntemin veri kümesinde yüksek ihtimalle doğru bulduğu düşünülen oranlarda eşleşen yüzlerle tekrar arama yapılmayan versiyonu kullanıldığında, sırasıyla%3.38, %3.78, %3.78, %3.78, %96.2, %0.2, %21.8, %98.8, %3.78, %98.8, %2.34 R-1, R-10, R-50, FNIR, FPIR, SELECTIVITY, FNIR-ALL, TPIR, Önerilen FNIR ve RN-T değerleri tespit edilmiştir. Tüm sonuçlar Tablo 4.7'de gösterilmiştir.

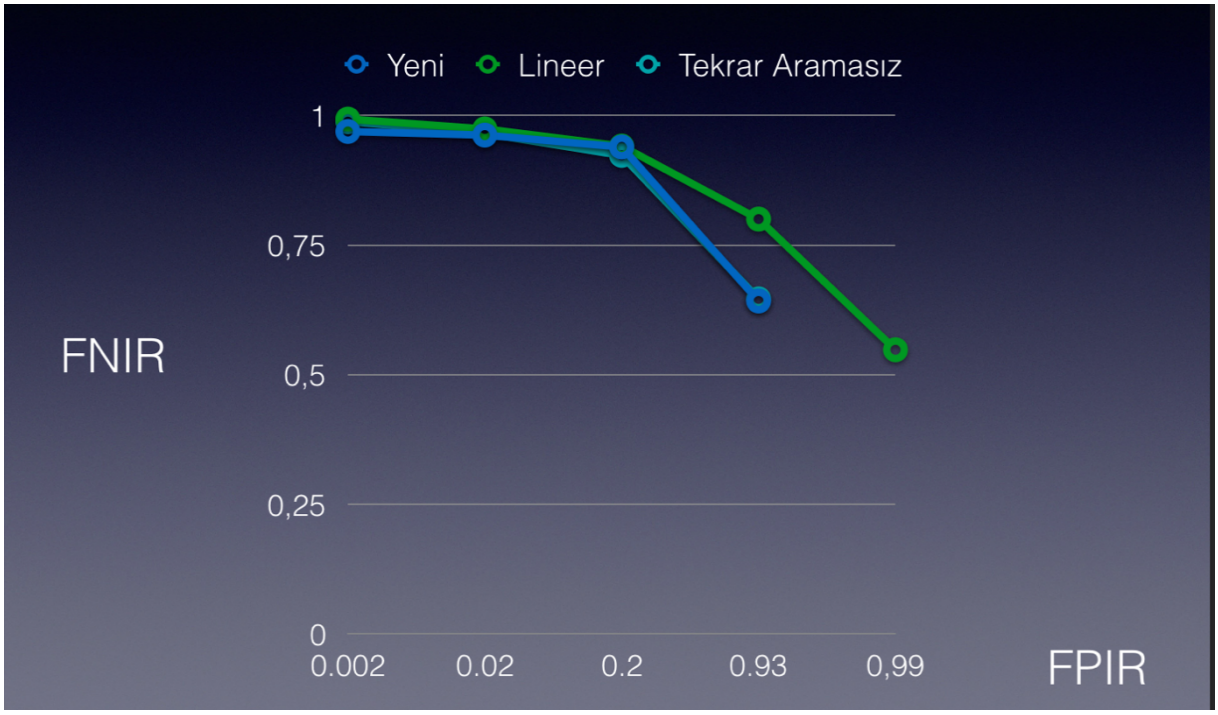
Bulunan değerlerle çizilen IET grafiği **Şekil 4.4'te gösterilmiştir**:



Şekil 4.4 IET Grafiği.

Şekil 4.4'te görüldüğü şekilde sol alta yakın mavi renkteki algoritma yeni yöntemin sonuçlarını içermektedir, aynı hata değerlerinde daha az yanlış negatif sonuç kaçırdığı için açık küme problemlerinde daha iyi sonuçların tespit edildiği söylenebilir.

Algoritmaların sadece tek bir doğruyu değil oranlı olarak ne kadar yanlış pozitif kaçırdığına göre oranlandırılarak oluşturulan Önerilen FNIR değerleriyle çizilmiş grafik Şekil 4.5'te görülebilir.



Şekil 4.5 Önerilen FNIR değeri ile DET Grafiği.

Şekil 4.5'te görülebileceği üzere daha fazla tahmin listesi elemanı içeren FPIR 0.2-0.9 arası değerlerde fark artmıştır. Fakat daha az sayıda tahmin listesi elemanı içeren düşük FPIR değerlerinde fark azalmıştır. Lineer aramaya ait ölçümlerde 0,99 değerinin bulunabilmesi ama yeni yöntemde bulunamamasının sebebi önerilen yöntemin en fazla T 0.6 eşik değerini kabul edebilmesidir. FPIR 0.99 sonucunu elde edebilmek için gereken T eşik değerini yükseltebilmenin tek yolu veri kümesini üretirken 0.6 eşik değerinden daha yüksek bir eşik değeri kullanıp o eşik değeriyle aramaktır.

Yeni yöntemin toplam çalışma süresi 17 saat 22 dakikayken aynı özellikte başka bir bilgisayarda lineer arama yönteminin çalışma süresi 19 saat 26 dakika olarak ölçülmüştür.

Yeni yönteme ek olarak galeride kesin olarak var olduğu bilinen diğer fotoğraflarla tekrar arama yapmayı engelleyen başka bir yöntem daha kullanılmıştır. O yöntemin çalışma süresi ise 6 saat 8 dakika olarak ölçülmüştür. Diğer yöntemlerle karşılaştırıldığında oldukça yüksek hızda çalışması avantajının yanında yeni yönteme oldukça yakın performans elde etmiştir.

5. TARTIŞMA VE SONUÇ

Lineer arama algoritmasında daha geniş bir uzay tarandığından sıralamalarda yanlış pozitifler doğru pozitiflerin önüne geçebilmektedir. Sunulan yeni yöntem log(N) karmaşıklığında çalıştığından lineer aramadan çok daha hızlı çalışmakta ve FNIR@FPIR skorlarının gösterdiği gibi lineer arama karakteristiğine benzer karakteristik göstermektedir.

Yeni önerilen yöntemdeki eşikleme doğrudan veri yapısında eşleşen ana katmanlar tarafından belirlendiği için listede bulunma ihtimali düşük doğru pozitifler listeye otomatik şekilde alınmamakta ve onun önüne geçebilecek yanlış pozitiflerle hiç arama yapılmamaktadır.

Yeni önerilen yöntem O(logN) karmaşıklığıyla veri eklemekte ve aramaktayken, lineer arama O(N) karmaşıklığıyla veriyi aramakta fakat O(1) karmaşıklıkta yeni verileri şablonlara eklemektedir.

Yeni önerilen yöntem hızlı arama yapabildiği için bir atomik arama için birden çok atomik arama yapılabilmektedir. Bu da veri kümesinde bulunan veriyle tekrar arama için vakit kazandırmakta ve tahmin listesini iyileştirmektedir. Tekrar arama yönteminin uygulandığı versiyonlarda düşük FPIR değerleri için performans artışı gözlenmiştir.

Lineer arama yönteminde eşik değerini sonsuza götürmek mümkündür. Yeni önerilen yöntemde veri yapısının doğası gereği aranacak T değerine uygun olarak veri yapısını kurgulama gereksinimi bulunmaktadır. O yüzden bir maksimum T değeri seçimi gerekmektedir. Yöntem o değerin üstünde tahmin listesi döndürememektedir. Bu problem IET grafiğinin çizimini de sınırlamaktadır. IET grafiğinin çizilmesi için farklı FPIR değerlerinde FNIR değerlerini raporlamak gerekmektedir. Eşik değeri gereksinimi olduğu ve bu değer sabit olduğu için IET grafiği FNIR=0 noktasına kadar çizilememektedir. Bu N tane fotoğraf bulunan L tahmin listesine sahip algoritmanın L=N noktası için çalışması için önceden kurgulanması anlamına gelmektedir. Çalışma kapsamında T eşik değeri olarak iki yüz arasında öklit uzaklık değeri olan 0.6 değeri kullanılmıştır. Her ne kadar T eşik değerinin üzerinde bir arama yapılabilecek olsa da bu arama yeni yöntemin lineer arama şeklinde davranmasına yol açmaktadır.

NIST, FRVT'yi kanun uygulayıcılara uygun tasarladığı için FNIR ve TPIR değerlerinin hesaplanışı ve yanlış negatif değerlerin kapsamı kanun uygulayıcıların gereksinime göre düzenlenmiştir. Senaryoların çoğunda kimliğe ait sadece bir fotoğrafın bulunması sorunu

çözdüğü için diğer fotoğrafların bulunamayışı göz ardı edilmiştir. Söz konusu veri tabanlarında aranan kimliğin birden çok fotoğrafı bulunuyorsa hangi oranda kaçırdığı veya hangi oranda bulduğuna bakılmamaktadır. Bu durum göz önüne alındığında algoritmaların değerlendirilişindeki FNIR skorunun standart sapması bakış açısına göre değişmektedir. Doğru pozitif sonuçlarda sonuçların hepsinin ne kadar üst sırada geldiği ve yanlış negatif sonuçlarda sonuçların kaçının arama listesine alınamadığı da önemlidir. Tüm kimliklerin sadece bir fotoğrafa sahip olduğu veri tabanlarında böyle bir sorun görülmemektedir. Fakat kimliklerin bir veya birden çok fotoğrafa sahip olduğu durumlarda FNIR ve TPIR skorlarını belirleyen bazı kimlikler baskın olmaktadır. Bu durum algoritma skorlarının birbirine benzemesine veya yaklaşmasına sebep olmaktadır. Skorlar arası farklılıkların nicel değerlerinin gerçek dünya uygulamalarındaki etkisi bu sebeple doğru ölçülemeyebilir. Örneğin galeride 6 fotoğrafı bulunan bir kimlik aramasının bir algoritma tarafından 1 fotoğrafıyla eşleşmesi FNIR ve TPIR skorlarını doğrudan galeride 5 fotoğrafıyla da eşleyen algoritmayla eşitlemektedir. Hassas T değerlerinin sonuçları oldukça değiştirdiği bir problem uzayında problemi yeteceği kadar veriyle ölçmek yeterli olmayabilir. Toleranslı veya yüzdeli bir FNIR skoru hesaplanışı algoritmaların kendi aralarında daha isabetli değerlendirebilmesini sağlayacaktır. Bu sebeple önerilen yeni bir FNIR skoru geliştirilmiştir.

Önerilen Yeni FNIR Skoru: Bir kimliğe ait N tane fotoğrafın olduğu veri tabanında sadece 1 eşleşmeye karşın N-1 eşleşmenin de aynı FNIR sonucunu almasının önüne geçmek için her arama için yanlış negatif kaçırma yüzdesi hesaplanır. Bu kaçırma yüzdesi eşleşmeyen kimlik fotoğrafı sayısı / veri tabanında kimliğe ait toplam fotoğraf sayısı şeklindedir. Kaçırma yüzdelerinin ortalaması NIST tarafından belirlenen FNIR değeri ile sadece 1 tane yanlış negatif sonuç ile fotoğraf kaçırmış FNIR1 değeri arasında yeni bir değer belirlemek için Denklem (5.1)'de gösterilen formül kullanılarak hesaplanmıştır:

$$\text{Önerilen FNIR} = \text{FNIR} + ((\text{FNIR1} - \text{FNIR}) * \text{kaçırma yüzdesi}) \quad (5.1)$$

Önerilen yeni yöntem yüz tanıma sistemlerinin bugüne kadar RAM bağımlı büyük tekil bilgisayarlardan ölçeklenebilir dağıtık sistemlere geçmesi için kullanılacak yöntemlerden bir tanesidir. Kesin olmayan komşu aramasını veri modelinden ve veriden bağımsız ölçekleyebilmeyi sağlayan bu yöntem arama hızlarını logaritmik düzeyde artırmaktadır. Veri ekleme için yine logaritmik bir yavaşlama sağlasa da gerçekleştiriminin kolay olması veriyi kopyalamaması, yakın komşu araması sistemlerinde bulunan çok yer kaplayan indeks sorununu çözmesi açısından özgündür. Verinin en güncel halinin hep aranabilir olması açısından yapısı

gereği indeksli bu model bellek ya da disk bağımlı yakın komşu araması çözümlerine bir alternatif sunmaktadır.

Önerilen yeni RN-T skoru ise tahmin listesinin yüksek sıralarında doğru pozitiflerin sayısını göz önüne alarak hesaplanan bir skordur. CMC grafiği doğru pozitif oranlarını hata oranlarını göz önüne almadan raporlarken algoritmanın daha iyi performans getirdiği atomik aramaların daha yüksek puanlaması amacıyla sunulmuştur. Bununla ilgili sonuçlar Şekil 4.5'te görülebilir. Tüm fotoğrafların tahmin listesine alındığı daha düşük yanlış negatif skora sahip yeni algoritma ilk fotoğrafı bulma konusunda daha kötü performans gösterirken ilk 20 sırada bir kimliğin galerideki tüm fotoğraflarını getirme konusunda daha iyi performans göstermiştir.

Yeni yöntem düşük FPIR değerlerinde daha iyi doğru pozitif sonuçlar vermektedir ve sıralama da daha iyi çıkmıştır. Yüksek FPIR değerlerinde doğru pozitif oranlarında %10'a yakın performans kaybı yaşanmıştır. Doğru pozitif sonuçların hepsinin tahmin listesine girebilmesi açısından yeni yöntem daha iyi sonuç vermiştir.

Yeni yöntemdeki iki farklı seçenekte FPIR=0.2 değeri dışındaki şartlarda tekrar arama algoritması daha iyi çalışmıştır. Yeni yöntemin açık küme problemlerinde sadece hızı artırdığı değil hata oranlarını da azalttığı, kapalı küme problemlerinde de yüksek FPIR değerlerinde lineer aramaya %10 kayıpla yaklaştığı görülmüştür. Yeni yöntemin geliştirilişi sırasında aynı kimliğin farklı fotoğraflarına ait tahmin listelerini sıralı birleştirmek yerine üzerinde istatistiki iyileştirme yapmak daha kötü sonuçlar vermiştir. FPIR 0.02 ve 0.9 aralığında daha fazla deney yapılarak daha derinlemesine incelenmesi gelecekteki araştırma konusu olabilir. Yüksek ve alçak FPIR değerlerinde yöntem iyi sonuç verirken ortalama FPIR değerlerinde tekrar arama algoritması hata oranlarını artırmıştır. SELECTIVITY değerleri ise yeni algoritmada tahmin listesinin %40'a varan oranda kısaltıldığını göstermektedir. Bu kısalmaya rağmen doğru pozitif sonuçlar sadece %10 azalmıştır.

Doğru pozitif sonuçların hepsini sıralamada yükseltmesi açısından yeni yöntem lineer aramaya göre daha iyi sonuç vermiştir.

Yeni yöntem düşük ve yüksek FPIR değerlerinde daha düşük FNIR değeri üretmiştir. Tekrar aramasız yeni yöntem FPIR 0.2 değerlerine yakın değerlerde tekrar aramalı yeni yönteme göre daha düşük FNIR değerine sahip olmasıyla daha yüksek başarı göstermiştir.

Önerilen yeni yöntemin bir veri kümesindeki tüm doğru pozitifleri açısından, 3 kat daha hızlı çalışması açısından, düşük ve yüksek FPIR değerleri üretecek eşik değerlerinde avantajlı olduğu fakat orta düzey FPIR değerlerini üretecek eşik değerlerinde en az 1 doğru pozitif getirmesi açısından performans kaybı ortaya çıkardığı söylenebilir.

Gelecekteki çalışmaların orta düzey FPIR değerlerinde algoritmaların daha detaylı karşılaştırılması ve sonuçlarının iyileştirilmesi, aynı kimliğin fotoğraflarına ait tahmin listelerini doğrudan birleştirmek yerine üzerinde uygulanabilecek istatistik fonksiyonlarıyla daha iyi hale getirilmesi üzerine geliştirilmelerin yapılması düşünülmektedir.

KAYNAKLAR

[1]. Mei Wang, Weihong Deng, 2018, *Deep Face Recognition: A Survey*, https://arxiv.org/abs/1804.06655

[2]. Gunther, Manuel Cruz ve Steve Rudd, Ethan ve M. Boult, Terrance E., 2017, *Toward Open-Set Face Recognition*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, ISBN: 9781538607336

[3]. Yan, J., Lei, Z., ... Li, S.Z., 2011. *Towards incremental and large scale face recognition*, in: 2011 International Joint Conference on Biometrics, IJCB 2011. doi:10.1109/IJCB.2011.6117583

[4] Vareto, R., Silva, S., ... Schwartz, W.R., 2018. Towards open-set face recognition using hashing functions, in: IEEE International Joint Conference on Biometrics, IJCB 2017. Institute of Electrical and Electronics Engineers Inc., pp. 634–641. doi:10.1109/BTAS.2017.8272751

[5]. Nech, A., Kemelmacher-Shlizerman, I., 2017. Level playing field for million scale face recognition, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. Institute of Electrical and Electronics Engineers Inc., pp. 3406–3415. doi:10.1109/CVPR.2017.363

[6]. Das, A., Mitra, S., Parua, S., 2011. Hierarchical clustering on texture statistics for search space reduction of large scale face recognition, in: ICIIP 2011 - Proceedings: 2011 International Conference on Image Information Processing. doi:10.1109/ICIIP.2011.6108924

[7]. Whitelam, C., Taborsky, E., ... Grother, P., 2017. IARPA Janus Benchmark-B Face Dataset, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, pp. 592–600. doi:10.1109/CVPRW.2017.87

[8] Klare, B.F., Klein, B., ... Jain, A.K., 2015. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 1931–1939. doi:10.1109/CVPR.2015.7298803

[9] Whitelam, C., Taborsky, E., ... Grother, P., 2017. IARPA Janus Benchmark-B Face Dataset, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, pp. 592–600. doi:10.1109/CVPRW.2017.87

[10] Parkhi, O. M. 2015, Features and methods for improving large scale face recognition, Thesis (PhD), Department of Engineering Science Oxford University.

[11] Kafai, M., Eshghi, K., Bhanu, B., 2014. Discrete cosine transform locality-sensitive hashes for face retrieval. IEEE Transactions on Multimedia 16, 1090–1103. doi:10.1109/TMM.2014.2305633

[12] Park, U., Jain, A.K., 2010. Face matching and retrieval using soft biometrics. IEEE Transactions on Information Forensics and Security 5, 406–415. doi:10.1109/TIFS.2010.2049842

[13] Wang, D., Otto, C., Jain, A.K., 2017. Face Search at Scale. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1122–1136. doi:10.1109/TPAMI.2016.2582166

[14] Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2001.990517

[15] Albiol, A., Monzo, D., ... Albiol, A., 2008. Face recognition using HOG-EBGM. Pattern Recognition Letters 29, 1537–1543. doi:10.1016/j.patrec.2008.03.017

[16] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005. pp. 886–893. doi:10.1109/CVPR.2005.177

[17] Kazemi, V., Sullivan, J., 2014. One millisecond face alignment with an ensemble of regression trees, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 1867–1874. doi:10.1109/CVPR.2014.241

[18] H.-W. Ng, S. Winkler. A data-driven approach to cleaning large face datasets. Proc. IEEE International Conference on Image Processing (ICIP), Paris, France, Oct. 27-30, 2014

[19] O. M. Parkhi, A. Vedaldi, A. Zisserman Deep Face Recognition British Machine Vision Conference, 2015.

[20] Gary B. Huang, Marwan Mattar, Tamara Berg, Eric Learned-Miller., 2018 Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Erik Learned-Miller and Andras Ferencz and Frédéric Jurie,

[21] Goldstein, J., Platt, J.C., Burges, C.J.C., 2005. Redundant bit vectors for quickly searching high-dimensional regions, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 137–158. doi:10.1007/11559887_9

[22] Wu, Z., Ke, Q., ... Shum, H.Y., 2011. Scalable face image retrieval with identity-based quantization and multireference reranking. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 1991–2001. doi:10.1109/TPAMI.2011.111

[23] Jégou, H., Douze, M., Schmid, C., 2011. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 117–128. doi:10.1109/TPAMI.2010.57

[24] Ponomarenko, A., Avrelin, N., 2014. Comparative Analysis of Data Structures for Approximate Nearest Neighbor Search. Data Analytics 125–130.

[25] Terasawa, K., Tanaka, Y., 2007. Spherical LSH for approximate nearest neighbor search on unit hypersphere, in: Lecture Notes in Computer Science (Including Subseries Lecture

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 27–38. doi:10.1007/978-3-540-73951-7_4

[26] Andoni, A., Indyk, P., 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS. pp. 459–468. doi:10.1109/FOCS.2006.49

[27] ISO/IEC 19795-1 :2006, Information Technology - Biometric Performance Testing and Reporting - Part I : Principles and framework.

[28] Grother, P., Ngan, M., 2014. Face Recognition Vendor Test (FRVT) Performance of Face Identification Algorithms. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.IR.8009

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Özgür Özkan
Doğum Yeri	ÇANKAYA
Doğum Tarihi	09.08.1993
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	+905416056632
E-Posta Adresi	ozgurozkan123@gmail.com
Web Adresi	https://www.linkedin.com/in/oscar-ozkan/



Eğitim Bilgileri	
Lisans	
Üniversite	Hacettepe Üniversitesi
Fakülte	Mühendislik
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	2015

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi-Cerrahpaşa
Enstitü Adı	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Tezli Yüksek Lisans

Makale ve Bildiriler