

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**UYDU GÖRÜNTÜLERİNDE HAVAALANLARININ
TESPİT EDİLMESİ**

Ümit BUDAK

Doktora Tezi
Anabilim Dalı: Elektrik-Elektronik Mühendisliği
Teknolojileri
Danışman: Prof. Dr. Abdulkadir ŞENGÜR

EKİM-2017

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYDU GÖRÜNTÜLERİNDE HAVAALANLARININ TESPİT EDİLMESİ

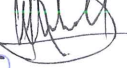
DOKTORA TEZİ

Ümit BUDAK

(132138202)

Tezin Enstitüye Verildiği Tarih: 18 Eylül 2017
Tezin Savunulduğu Tarih: 17 Ekim 2017

Tez Danışmanı: Prof. Dr. Abdulkadir ŞENGÜR (F.Ü) 

Diğer Jüri Üyeleri: Doç. Dr. Melih Cevdet İNCE (F.Ü) 

Doç. Dr. Davut HANBAY (İ.Ü) 

Doç. Dr. Murat KARABATAK (F.Ü) 

Yrd. Doç. Dr. Ömer Faruk ALÇİN (B.Ü) 

EKİM-2017

ÖNSÖZ

Tez çalışmam süresince bilgi ve tecrübelerinden faydalandığım ve desteğini esirgemeyen danışman hocam Sayın Prof. Dr. Abdulkadir ŞENGÜR'e, ayrıca değerli görüş ve fikirlerini benimle paylaşan, akademik anlamda yol gösteren ODTÜ Elektrik - Elektronik Mühendisliği Öğretim Üyesi Sayın Prof. Dr. Uğur HALICI'ya, Bingöl Üniversitesi Elektrik - Elektronik Mühendisliği Öğretim Üyesi Yrd. Doç. Dr. Ömer Faruk ALÇİN'e ve Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Öğretim Üyesi Doç. Dr. Murat KARABATAK'a sonsuz teşekkürlerimi sunarım.

Son olarak hayatımın her anında yanımda olan ve bu süreçte ihmal etmiş olduğum eşime, biricik kızlarım Ela ve İdil'e göstermiş oldukları sabırdan dolayı şükranlarımı sunarım.

Ümit BUDAK
ELAZIĞ-2017

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	IX
KISALTMALAR LİSTESİ	X
SEMBOLLER LİSTESİ	XI
1. GİRİŞ	1
1.1. Havaalanının Yapısı.....	1
1.2. Literatür Taraması	2
1.3. Tezin Kapsamı	5
1.4. Tezin Organizasyonu	7
2. KENAR VE DOĞRU PARÇASI ALGILAYICILARI.....	9
2.1. Roberts, Sobel ve Prewitt Kenar Algılayıcıları	10
2.2. LoG Kenar Algılayıcısı.....	11
2.3. Canny Kenar Algılayıcısı	12
2.4. Doğru Parçası Algılayıcısı.....	15
3. ÖNERİLEN HAVAALANI TESPİT YÖNTEMİ	18
3.1. Havaalanı Aday Bölgelerin Tespiti İçin Geliştirilmiş Doğru Parçası Algılayıcısı.....	18
3.2. Ölçekten Bağımsız Özellik Dönüşümü	24
3.2.1. Ölçek-Uzay Yerel Maksimum ve Minimumların Tespiti.....	24
3.2.2. Doğru Kilit Noktaların Lokalizasyonu	29
3.2.3. Yönelim Atama.....	31
3.2.4. Kilit Nokta Tanımlama	33
3.3. Fisher Vektör Kodlama.....	35
3.3.1. Örnek Bir FV Uygulaması.....	36
3.4. Destek Vektör Makineleri.....	39
3.5. Evrişimsel Sinir Ağları	41
3.5.1. ESA Yapısı	42

3.5.1.1.	Evrişim Katmanı.....	43
3.5.1.2.	Havuzlama Katmanı	44
3.5.1.2.1.	Alt Örnekleme Havuzlama	45
3.5.1.2.2.	Maksimum Havuzlama.....	46
3.5.1.3.	Aktivasyon Fonksiyonu.....	47
3.5.1.3.1.	Lojistik Sigmoid Fonksiyonu	47
3.5.1.3.2.	Hiperbolik Tanjant Fonksiyonu.....	48
3.5.1.3.3.	Düzeltilmiş Doğru Birimi.....	49
3.5.1.4.	Bırakma.....	50
3.5.1.5.	Kayıp Fonksiyonu.....	52
3.5.1.5.1.	Öklid Fonksiyonu	52
3.5.1.5.2.	Softmax Fonksiyonu.....	52
3.6.	Performans Ölçütleri.....	53
3.6.1.	Karışıklık Matrisi.....	53
3.6.1.	k-Katlı Çapraz Doğrulama.....	55
4.	DENEYSEL SONUÇLAR	57
5.	SONUÇLAR.....	68
	KAYNAKLAR	70
	EKLER	79
	ÖZGEÇMİŞ	86

ÖZET

Son yıllarda uydu görüntülerinden otomatik havaalanı tespiti, askeri ve sivil havacılık alanlarındaki öneminden dolayı popüler bir konu haline gelmiştir. Ancak binalar, dağlar, nehirler ve yollar gibi karmaşık arka plana sahip olan bu görüntüler, havaalanlarının tespitini zorlaştırmaktadır. Ayrıca uydu görüntüleri büyük boyuta sahip olduğundan, hesaplama maliyetini azaltmak için algoritmanın karmaşıklığını da hesaba katmak oldukça önemlidir.

Bu tez çalışmasında, uydu görüntülerinden otomatik havaalanı tespiti için iki farklı yaklaşım önerilmiştir.

Birinci yöntemde, öncelikle potansiyel havaalanı aday bölgelerin tespiti için Doğru Parçası Algılayıcısının (DPA) geliştirdiğimiz versiyonu kullanılmıştır. Daha sonra bu aday bölgelerden Ölçekten Bağımsız Özellik Dönüşümü (ÖBÖD) ile etkili özellikler çıkarılmış ve bu özellikler Fisher Vektörü (FV) beslemiştir. FV’de görsel sözlük olarak Gauss Karışım Modeli (GKM) kullanılmıştır. Tüm Gauss’ların birinci ve ikinci dereceden kısmi türevleri birleştirilerek FV’ler elde edilmiştir. Özellik vektörü olarak FV kullanılmış ve sınıflandırıcı aşamasında Destek Vektör Makinesi (DVM) yetkilendirilmiştir. Önerilen yöntemin performansını değerlendirmek için doğruluk, duyarlılık ve özgüllük kriterleri kullanılmış ve %94,6’lık doğruluk elde edilmiştir. Önerilen yöntemin başarımlı üstünlüğü, literatürde daha önceden önerilmiş diğer çalışmalar ile karşılaştırılarak kanıtlanmıştır. Ayrıca havaalanı aday bölgelerin tespiti için geliştirdiğimiz DPA’nın, diğer yöntemlere göre daha az hesaplama maliyetine sahip olduğu karşılaştırmalı olarak doğrulanmıştır.

İkinci yöntemde ise, uydu görüntülerinden havaalanlarının otomatik tespiti için hem özellik çıkarımı hem de sınıflandırıcı yapısını birlikte barındıran derin Evrişimsel Sinir Ağlarının (ESA) kullanımı araştırılmıştır. Öncelikle geliştirdiğimiz DPA yöntemi ile aday havaalanı bölgeleri tespit edilmiştir. Bu bölgeler, önerilen birinci yöntemdeki aday bölgeler ile aynıdır ve ESA mimarisini eğitmek için kullanılmıştır. ESA modeli, beş evrişim, üç havuzlama ve üç tam bağlı katmandan oluşmaktadır. Verimli bir mimari oluşturmak için normalleştirme ve bırakma katmanları da kullanılmıştır. Aşırı uyumu azaltmak için eğitim verisine ait adaylar, yapay olarak artırılmıştır. ESA performansını değerlendirmek için yine doğruluk, duyarlılık ve özgüllük kriterleri kullanılmıştır. Önerilen yöntem %95,21’lik doğruluğa sahiptir ve birinci öneriye göre yaklaşık %1’lik bir başarımlı artışı sağlanmıştır.

Anahtar Kelimeler: Havaalanı Tespiti, Uzaktan Algılama Görüntüleri, Doğru Parçası Algılayıcısı, Ölçekten Bağımsız Özellik Dönüşümü, Fisher Vektör, Destek Vektör Makineleri, Evrişimsel Sinir Ağları.

SUMMARY

Detection of Airport in Satellite Images

In recent years, automatic detection of airport from satellite images has become a popular topic due to its importance in military and civil aviation fields. However, these images which have complex backgrounds such as buildings, mountains, rivers and roads, make it difficult to identify airports. In addition, since satellite images have a large size, it is crucially important to consider the complexity of the algorithm to the account in order to reduce the computational cost.

In this thesis, two different approaches are proposed for automatic airport detection from satellite images.

In the first method, initially the version of Line Segment Detector (LSD) we developed is used for the detection of potential airport candidate regions. Then efficient features are extracted from these candidate regions with Scale Invariant Feature Transform (SIFT). This features are fed into Fisher Vector (FV). In FV, Gauss Mixture Model (GMM) is used as visual dictionary. FV is obtained by combining first and second partial derivatives of all Gauss. FV is used as the feature vector and Support Vector Machine (DVM) is employed in the classifier stage. Accuracy, sensitivity and specificity criteria are used to evaluate the performance of the proposed method and yielded an accuracy of 94.6%. The superiority of the performance of the proposed first method has been proved by comparison with other studies previously suggested in the literature. We have also verified that the LSD we have developed for the detection of airport candidate regions has comparatively lower computational costs than other methods.

In the second method, the usage of deep Convolutional Neural Networks (CNNs), which contain both feature extraction and classifier structure has been investigated for the automatic detection of airports from satellite images. Firstly, candidate airport regions have been detected by using the LSD method we developed. These regions are the same as the candidate regions in the first proposed method and used to train CNN architecture. The CNN model consists of five convolutions, three pools and three fully connected layers. Normalization and dropout layers have also been used to create an efficient architecture. Candidates in training data have been artificially increased to reduce overfitting. Accuracy, sensitivity and specificity criteria were also used to evaluate CNN performance. The second proposed method has accuracy of 95.21% and an increase of about 1% was achieved according to the first proposal.

Key Words: Airport Detection, Remote Sensing Images, Line Segment Detector, Scale-Invariant Feature Transform, Fisher Vector, Support Vector Machines, Convolutional Neural Networks.

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Önerilen birinci havaalanı tespit yöntemine ait akış şeması.....	6
Şekil 1.2. Önerilen ikinci havaalanı tespit yöntemine ait akış şeması.....	7
Şekil 2.1. Kenarların belirli açılarla ilişkilendirilmesi.....	13
Şekil 2.2. Kenar bulma sonuçları; (a) Orijinal görüntü, (b) Roberts operatörü, (c) Sobel operatörü, (d) Prewitt operatörü, (e) LoG operatörü, (f) Canny operatörü.....	14
Şekil 2.3. (a) Orijinal gri seviyeli görüntü, (b) seviye-doğru alanı, (c) doğru destek bölgesi.....	15
Şekil 2.4. Doğru parçası algılayıcısının sonucu.....	17
Şekil 3.1. Doğru parçalarının eliminasyonu, (a) DPA'nın çıkışı, (b) Eliminasyon işlemi sonrası	19
Şekil 3.2. (a) Doğru parçası etiketleme işleminin gösterimi, (b) Havaalanı görüntüsünde doğru parçası etiketleme	19
Şekil 3.3. İki paralel L_1 ve L_2 doğru parçaları ve onlar arasındaki D_p dik mesafe.....	20
Şekil 3.4. İki doğru parçasının bağlantısı	21
Şekil 3.5. UAG'de bağlantı prosedür sonucu, (a) Adım 4'ten önce, (b) Adım 4'ten sonra.....	22
Şekil 3.6. Doğru parçası azaltma, (a) Adım 5'ten önce, (b) Adım 5'ten sonra	22
Şekil 3.7. Nihai doğru parçası eliminasyonu (a) Adım 6'dan önce, (b) Adım 6'dan sonra.....	23
Şekil 3.8. Ölçek uzayının her oktavında farklı Gauss ölçeği ile yumuşatılmış görüntü setinin elde edilmesi (sol sütun), ve bitişik Gauss görüntülerinin farkı (sağ sütun)	27
Şekil 3.9. Bir oktav için aday bölgelerden Gauss farklarının elde edilmesi	28
Şekil 3.10. Yerel maksimum ve minimumların tespit edilmesi.....	28
Şekil 3.11. Elimine işlemlerinden sonra havaalanı aday bölge üzerinde bulunan kilit noktalar	32
Şekil 3.12. (a) Görüntünün gradyanları, (b) anahtar nokta tanımlayıcı	33
Şekil 3.13. Farklı ölçekteki görüntülerin eşleştirme işlemi	34
Şekil 3.14. Farklı ölçek ve yöndeki görüntülerin eşleştirme işlemi.....	35
Şekil 3.15. İki sınıflı veri kümesi için en uygun hiper düzlemin yerinin gösterimi	40

Şekil 3.16. ESA mimarisinin akış şeması	43
Şekil 3.17. Ağırlığı paylaşılan evrişim katmanları ile tam bağlı katmanın karşılaştırılması.....	43
Şekil 3.18. Ortalama havuzlama örneği.....	45
Şekil 3.19. Maksimum havuzlama örneği.....	46
Şekil 3.20. $\text{sigm}(x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için)	47
Şekil 3.21. $\text{tanh}(x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için).....	48
Şekil 3.22. $\text{max}(0,x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için)	49
Şekil 3.23. Basit bir bırakma örneği. Sol: iki gizli katmana sahip standart bir sinir ağı. Sağ: Sol taraftaki ağa bırakma uygulanarak üretilen inceltilmiş bir ağ örneği.	50
Şekil 3.24. Sol: eğitim sürecinde bir birim. Sağ: Test sürecinde bir birim. Eğitim sürecindeki birim, olasılık p ile sunulur ve eğitilmiş w ağırlığı ile bir sonraki katmanda birimlere bağlanır. Test sürecinde, Birim her zaman mevcuttur ve ağırlıklar p ile çarpılır. Test sürecinin çıktısı, eğitim sürecinin beklenen çıktısıyla aynıdır	51
Şekil 3.25. Karışıklık matrisi	54
Şekil 3.26. 5-katlı çapraz doğrulama örneği	55
Şekil 4.1. Havaalanı tespit örneği	59
Şekil 4.2. Havaalanı bölgeleri	62
Şekil 4.3. Havaalanı olmayan bölgeler	62
Şekil 4.4. Veri artırma işlemi. Birinci satır sırasıyla: orijinal görüntü, X, Y ve XY eksenine göre çevrilmiş görüntü. İkinci satır sırasıyla: 90° döndürülmüş, Y eksenine çevrilmiş, -90° döndürülmüş ve Y eksenine çevrilmiş görüntü.....	63
Şekil 4.5. Havaalanı tespiti için kullanılan ESA mimarisi	64
Şekil 4.6. Birinci evrişim katmanı tarafından öğrenilen $11 \times 11 \times 3$ boyutunda 64 adet evrişim çekirdeği	65
Şekil 4.7. Başarılı bir şekilde tespit edilmiş havaalanı görüntüleri; (a) tespit edilmiş havaalanı aday görüntüleri, (b) doğrulanmış havaalanları	66

TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Roberts, Sobel ve Prewitt evrişim maskeleri.....	10
Tablo 2.2. Kenar algılayıcılarının yönelim açıları	10
Tablo 2.3. Farklı ölçeklerde LoG maske yapıları	11
Tablo 4.1. Geliştirilen DPA için seçilen parametreler	58
Tablo 4.2. Önerilen birinci yöntemin performansı	60
Tablo 4.3. Önerilen birinci yöntem ile literatürdeki diğer yöntemlerin karşılaştırılması ..	60
Tablo 4.4. Önerilen yöntemin işlem süresi bakımından karşılaştırılması.....	61
Tablo 4.5. Önerilen ikinci yöntemin performansı.....	67

KISALTMALAR LİSTESİ

DÇ	: Dikkat Çekerlik
DPA	: Doğru Parçası Algılayıcısı
DVM	: Destek Vektör Makineleri
ESA	: Evrişimsel Sinir Ağları
FV	: Fisher Vektör
GKM	: Gauss Karışım Modeli
GTGDÇ	: Grafik Tabanlı Görsel Dikkat Çekerlik
HDR	: Hiyerarşik Diskriminant Regresyon
LoG	: Gauss Laplace
ÖBÖD	: Ölçekten Bağımsız Özellik Dönüşümü
UAG	: Uzaktan Algılama Görüntüleri



SEMBOLLER LİSTESİ

$ G $	: Gradyan genliği
$\theta(x,y)$	: Yönelim açısı
$I(x,y)$	: Giriş görüntüsü
$L(x,y)$	: Görüntünün laplace'ı
D_p	: İki doğru arasındaki dik mesafe
d	: Bir doğru parçasının sonu ile diğer doğru parçasının başlangıç noktası arasındaki mesafe
H	: İkinci dereceden türevlerin hessian matrisi
G_λ^x	: Gradyan vektörü
u_λ	: Olasılık yoğunluk fonksiyonu
F_λ	: Olasılık yoğunluk fonksiyonunun bilgi matrisi
g_λ^x	: Fisher vektörü
γ_i	: Fonksiyonel marjin
w	: Ağırlık vektörü
h^k	: Evrişim maskesi
h^{k+1}	: Özellik haritası
b^k	: Bias
p_j	: Softmax fonksiyonu
$*$	: Evrişim operatörü

1. GİRİŞ

Havaalanları hem sivil hem de askeri açıdan önemli yapılardır. Sivil açıdan temel kargo ve yolcu taşımacılığı olarak hizmet vermektedirler. Böylece havaalanları finansal sebepler açısından ulusal ve küresel ekonomide önemli bir güç teşkil etmektedir.

Askeri havaalanları diğer bir deyişle hava üsleri, bir ulusun savunma güçlerinin havacılık dalı göz önünde bulundurulduğunda kritik stratejik hedeflerdir. Havaalanları sadece hava araçlarının kalkışı ya da bombardıman ve savaş ünitelerinin inişleri için değil, ayrıca stratejik hava nakliyesi, havadan yardım ve tıbbi tahliye olarak kullanıldığı için önem derecesi artmaktadır. Bu açıdan bakıldığında, savaş halindeyken zamanında askeri tedbirler almak için havaalanlarının otomatik tespiti hayati istihbarat sağlayabilmektedir.

Hem hesaplama dayalı donanım hem de doku tanımlama tekniklerindeki teknolojik gelişmeler, havaalanlarının tespitini ulaşılabılır bir amaç haline getirmiştir. Bununla birlikte kendi uydularına sahip olan ülkelerin sayısının artmasından dolayı sorun daha cazip hale gelmektedir. Bahsedilen bu sebepler, bu tezin motivasyonunu oluşturmaktadır.

1.1. Havaalanının Yapısı

Son yıllarda uydu görüntülerinden otomatik hedef tespiti önemli bir uygulama olarak karşımıza çıkmaktadır. Yüksek çözünürlüklü Uzaktan Algılama Görüntüleri (UAG), yollar, binalar, orman arazileri, araçlar gibi hedeflerin tespitinde sıkça kullanılmaktadır. Hem askeri hem de sivil uygulamalar için vazgeçilmez alanlar olarak düşünülen havaalanları da yine bu uygulamalar içinde önemli bir yer teşkil etmektedirler. Ancak havaalanlarının zeminleri karmaşık olma eğilimindedir ve genelde yollar, nehirler ve binaların bulunduğu şehir içi ya da şehirlerarasındaki bölgeler, havaalanı ile benzer özellikler gösterebilmektedir [1,2].

Sivil havaalanları genellikle bir ya da daha fazla pist, taksi yolları, yolcu ve yük terminalleri, bir kontrol kulesi, yolcu yürüyüş yolları ile park alanları gibi diğer küçük binalara sahiptir. Ayrıca planör ve deniz uçağı için özel olarak inşa edilmiş yapılar da sivil havaalanları olarak düşünülmektedir.

Askeri havaalanları da genellikle bir ya da birden fazla pist, taksi yolları, hangarlar, kışlalar, kontrol kulesi, savunma yapıları ile sığınak, yönetim binası, yangınla mücadele istasyonu ve mühimmat deposu gibi diğer küçük yapılara sahiptir. Jet uçakları için inşa

edilen askeri havaalanları uzun pistlere sahipken, nakliye uçakları için inşa edilmiş havaalanları da genellikle daha geniştir.

Temel olarak havaalanı pisti uçakların kalkış esnasında güvenli bir şekilde istenilen hıza ulaşması için gerekli mesafeyi sağlayan uzun bir şeritten oluşmaktadır. Bir havaalanında tekli pistler ya da paralel ve çaprazlama gibi birçok pist, çeşitli konfigürasyonda inşa edilebilmektedir.

1.2. Literatür Taraması

Önceki çalışmalarda havaalanı tespit problemleri için düşünülen yaklaşımlar iki grupta sınıflandırılabilirler [2]. Yaklaşımlardan biri kenar ya da doğru parçasına, diğeri ise doku ya da görüntü bölütlemeye dayalıdır [1,3-6]. Kenar ya da doğru parçalarını kullanan yaklaşımlar, havaalanı pistlerinin karakterize edilmesini sağlamaktadır.

Kou ve arkadaşları, Doğru Parçası Algılayıcısı (DPA) tabanlı bir havaalanı tespit yöntemi önermiştir [7]. Yöntem, öncelikle DPA'nın ürettiği parçalı doğruları birleştirmekte ve daha sonra havaalanları pistlerinin uzun paralel doğrulara sahip olması gerektiği gerçeğini kullanarak havaalanı bölgelerini tespit etmektedir.

Son zamanlarda etkili bir havaalanı tespiti için kenar ya da doğru parçasına dayalı yaklaşımların ve bölütleme tabanlı yaklaşımların birleştirildiği birçok yeni yöntem mevcuttur [8-10].

Kaynak [10]'da Zhu ve arkadaşları, havaalanı tespiti için iki-yönlü bir Dikkat Çekerlik (DÇ) yöntemi önermiştir. Buna göre UAG'lerde potansiyel havaalanı bölgelerini bulmak için hem DPA [11] hem de Grafik Tabanlı Görsel Dikkat Çekerlik (GTGDÇ) [12] yöntemlerini kullanmıştır.

Kısa ve eğri çizgilerin kaldırıldıktan sonra geriye kalan uzun düz çizgilerin potansiyel havaalanı bölgesi olarak kaydedildiği benzer bir çalışma da Qu ve arkadaşları tarafından yürütülmüştür [9]. Potansiyel bölgelerin sınıflandırılması da dokusal özellikler sayesinde elde edilmiştir.

Kaynak [8]'de havaalanı tespiti için Hough Dönüşümü ve GTGDÇ yöntemleri birleştirilmiştir. Yaklaşım Hough dönüşümü ile başladıktan sonra DÇ haritasını hesaplamak ve potansiyel havaalanı bölgelerini çıkarmak için geliştirilmiş GTGDÇ yöntemi kullanılmıştır. Son olarak Ölçekten Bağımsız Özellik Dönüşümü (ÖBÖD) özellikleri ve

Hiyerarşik Diskriminant Regresyon (HDR) ağacı sınıflandırıcısı kullanılarak havaalanları belirlenmiştir.

Tao ve arkadaşları da havaalanlarını ÖBÖD özellikleri ile karakterize etmiştir [13]. Bu yaklaşım, bölütlenmiş bölgeleri ve ÖBÖD özelliklerini birleştirmektedir. Son olarak havaalanları, ortalama, varyans ve diğer istatistiksel büyüklükler ile tespit edilmiştir.

Zuo ve arkadaşları [14] havaalanı tespiti için kızıl ötesi görüntülerde şablon eşleştirmeye dayalı yeni bir isabet-hedef konumlandırma algoritması önermişlerdir. Öncelikle havaalanı pisti ve arka plan arasındaki farkı büyütme için eş-oluşum matrisine dayalı homojenlik dönüşüm yöntemi kullanılmıştır. Dönüştürülmüş görüntü bölütlendikten sonra pistin tamamını eksiksiz oluşturmak için bölütlenmiş ikili görüntüye morfolojik filtre uygulanmıştır. Son olarak şablon eşleştirme yöntemi ile havaalanı pisti konumlandırılmıştır.

Wei ve arkadaşları ise, gece görüş kamerasından elde edilen kızıl ötesi görüntülerde otomatik olarak pist bölgesini bulan bir yöntem önermişlerdir [15]. Öncelikle görüntülerdeki doğru kısımlar, hızlı bir DPA ve geliştirilmiş doğru parçası bağlama yöntemleri kullanılarak çıkartılmıştır. Pistlerin yön, gradyan ve doğru parçası çiftinin genişliğine dayanan bazı kısıtlayıcı kuralları kullanarak ilgili bölgeler tespit edilmiştir. Daha sonra doku dağılımına dayalı bir uzunluk geri izleme tekniği, ilgili bölgeyi eksiksiz elde etmek için sunulmuştur. Son olarak ta öz-benzerlik ve bağlamsal bilgileri kullanarak havaalanı pistini tespit eden çeşitli karar kriterleri oluşturulmuştur.

Benzer bir çalışmada ise Dan ve arkadaşları, uydu görüntülerinden havaalanı tespiti için havaalanı pisti bölümlerinin geometrik özelliklerini sınırlamada kullanılan hem aşağıdan-yukarıya hem de yukarıdan-aşağıya DÇ tabanlı yeni bir yöntem önermişlerdir [16]. Havaalanı pistlerinin çevresinde paralel yapıya sahip özellikler mevcuttur ve uzunlukları belirli bir aralıktadır. Yakın paralellik kavramı, DPA ve bir bağlantı işlemi tamamlandıktan sonra eklenmiştir. Bu da, yukarıdan-aşağıya DÇ'yi elde etmek için tamamen havaalanı pistlerinin geometrik ilişkilerinden yararlanabilen önsel bir bilgi olarak kabul edilmiştir. Bu sırada geliştirilmiş GTGDÇ modeli aşağıdan-yukarıya DÇ'yi ayıklamak için kullanılmıştır. Aday bölgeler, bu çift yönlü sonuçları birleştirerek elde edilmiştir. Son olarak ÖBÖD özellikleri ve Destek Vektör Makinesi (DVM) sınıflandırıcısı, bölgede havaalanının olup olmadığını belirlemek için kullanılmıştır.

Yinwen ve arkadaşları [17], uydu görüntülerinde normalize edilmiş kenar görüntülerini elde etmek için Sobel operatörü kullanmış ve kenar görüntüsünde sezgisel bir tarama kullanılarak uygun düz doğrular çıkarılmıştır. Daha sonra düz doğruları bağlayarak paralel

doğrular çıkarılmıştır. Son olarak havaalanı pistinin gri alanın özelliklerine göre doğru bir şekilde tanımlama yapılarak havaalanı pisti tespit edilmiştir.

Wang ve Zhang [18], havaalanı pistlerini tespit etmek için, kenar belirleme ve vektör çizgi katmanındaki derinlik-öncelik aramayı birleştiren düz doğru çıkarma yöntemi önermişlerdir. Önerilen yöntem dört adımda gerçekleştirilmiştir. Birinci adımda kenarlar Canny operatörü ve vektörleştirme ile bulunmuştur. Canny operatörü optimum kenar belirleme olarak tasarlanabildiğinden dolayı tercih edilmiştir ve kenarlar gibi adım ve yansımaları belirlemede iyi sonuç vermektedir. Gri seviyeli giriş görüntüsü ile izlenen yoğunluk süreksizliklerin konumlarını gösteren bir çıkış görüntüsü elde edilmiştir. Bu işlemden sonra kenar takibi ile vektör katmanı olan kenar noktalar vektörize edilmiştir. İkinci adımda vektör olarak şekillendirilmiş kenar çizgileri sayesinde düz doğru araması gerçekleştirilmiştir. Bunu yapmak için tekrarlayan silinme, tek düğümlü yaylar ve kesişimler üzerindeki bölmeleri içeren yaylar arasındaki topoloji yeniden inşa edilmiştir. Üçüncü adımda düz doğrular, derinlik öncelikli arama stratejisi ile belirlenmiştir. Topolojinin tekrar inşa edilmesi ile birlikte her hattın başı ve son düğümü kolaylıkla elde edilebilmektedir. Eğer başlangıç köşesinden bitiş köşesine olan mesafe, önceden tanımlanmış bir eşik değerinden düşük ise orası düz bir doğru olarak görülür ve ayıklanabilmektedir. Dahası ilgili düz doğrular, belirli bir eşiğin üzerinde kalan uzunluklardır. Böylece minimum uzunluk eşiği bu kısa doğruların elenmesinde rol oynamaktadır. Düz doğruların aranmasında bazı yaylar, tek düz bir doğru olarak gruplandırılmış olmalıdır. Son adımda ise paralel düz doğrular arasındaki boşluklar etkili bağlantı adımlarıyla doldurularak pist bölgesi tespit edilmiştir.

Tripathi ve arkadaşları [19], basit ve etkili bir havaalanı pisti bulma algoritması önermişlerdir. Önerilen algoritma iki aşamadan oluşmaktadır. İlk aşama havaalanı yolunun bölütleme adımıdır. Bu aşamada yön bağımlı (anizotropik) difüzyon ve Frangi filtresi kullanılmıştır. Yön bağımlı difüzyon gürültü bağımsızlığı sağlar ve Frangi filtresi ise olası tüm pist adaylarını tespit eder. İlk aşamada üretilen yanlış adaylardan kaçınmak için, şekil ve renk tabanlı özellikler kullanılarak aday piksellerin ikinci aşamaya geçmesi sağlanmıştır.

Jinglin ve arkadaşları [20] ise, JPEG2000 sıkıştırılmış etki alanı için uzaktan algılama tabanlı havaalanı tespit yöntemi önermiştir. Havaalanının en temel özelliği belirli gri ton, uzunluk ve genişliğe sahip paralel pist olmasıdır. Sıkıştırılmış veriler daha az açma (dekompresyon) ve dalgacık tabanında işlenmiştir. Havaalanı pisti Hough Dönüşümü ile dalgacık katsayılarının LL bandında tespit edilmiştir ve paralel çizgilerin tanımlanması ile hava alanının yeri belirlenmiştir.

Aytekin ve arkadaşları, doku ve renk özelliklerini kullanarak havaalanı pistlerini yüksek doğrulukla belirlemişlerdir [1]. Yöntemde yazarlar, havaalanı pistlerinden ve havaalanı olmayan bölgelerden 32×32'lik örnekler almışlar ve daha sonra bu örneklerden 137 boyutlu bir özellik çıkarmışlardır. Özellikler Adaboost yöntemi ile sınıflandırılmıştır. Yöntem, test aşamasında 32×32'lik kayan bir pencere ile uydu görüntüsünü taramakta ve pistlerin yerini belirlemektedir.

Tang ve arkadaşları yine DPA ve doku özelliklerini birlikte kullanarak uydu görüntülerinde havaalanı tespit edebilen bir yaklaşım önermişlerdir [21]. Öncelikle çıkarılan havaalanı pist doğruları bulunur ve daha sonra pist çevresi bölgesinin doku özellikleri ile sınıflandırma işlemi tamamlanmaktadır.

Yukarıda bahsedilen literatür çalışmaları detaylı bir şekilde incelendiğinde, havaalanı tespit çalışmalarının genellikle DPA tabanlı bir aday çıkarma ve sonrasında aday bölgelerden elde edilen özelliklerin bir sınıflandırıcı ile doğrulanması esasına dayandığı görülmektedir.

1.3. Tezin Kapsamı

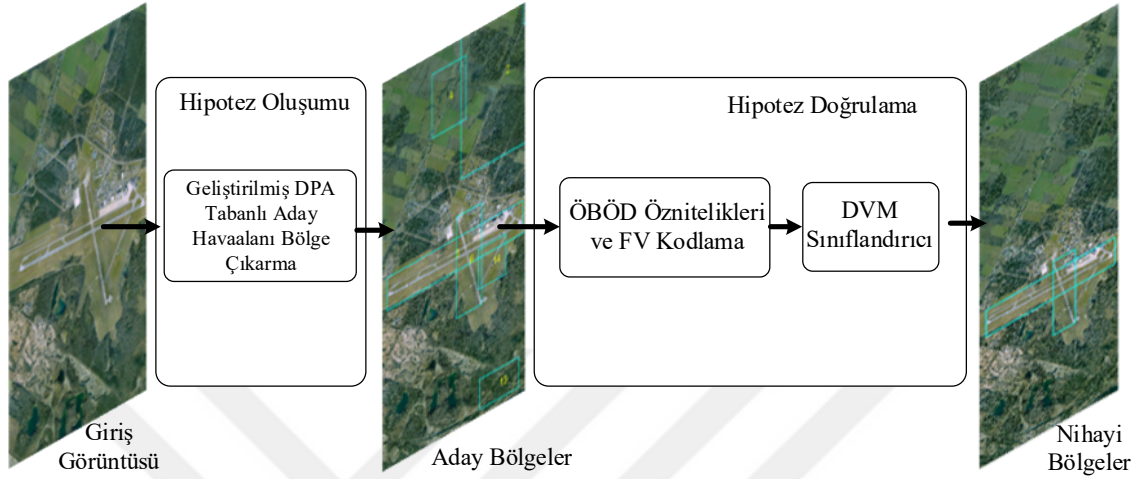
Bu tez çalışmasında UAG'lerden havaalanı tespiti için iki farklı yaklaşım önerilmiştir.

Birinci öneride, Şekil 1.1'de blok şeması verilen yöntem sunulmuştur. Önerilen yöntem iki temel aşamadan oluşmaktadır; 1-) Hipotez oluşum: Geliştirilmiş DPA algoritmasına dayanan aday havaalanı bölgeleri çıkarma, 2-) Hipotez doğrulama: Aday bölgelerin analizi için ÖBÖD özellikleri çıkarma, FV özellikleri kodlama ve DVM sınıflandırıcısı.

Hipotez oluşum aşamasında çıkarılan havaalanı aday bölgeleri, uzun ve paralel doğru parçalarını içermektedir. Bu aşamada UAG'lerdeki düz doğru parçalarını bulmak için DPA algoritması kullanılmıştır [11]. Ancak DPA algoritması doğrudan uygulandığında çeşitli uzunluklarda çok sayıda doğru parçası üretilmektedir. Genelde bu doğru parçaları havaalanlarının pisti boyunca tek bir doğru şeklinde üretilmemiştir. Bu bağlanmamış doğru parçaları, havaalanı pistlerinin tamamını kapsayacak şekilde belirtmek için yeterli değildir. Bu da havaalanı tespit uygulamaları için ciddi bir sorun teşkil etmektedir. Bu sorunu gidermek ve muhtemel havaalanı aday bölgelerini tespit etmek için Bölüm 3.1'de detaylı bir şekilde açıklanan DPA'nın geliştirdiğimiz versiyonu kullanılmıştır.

Hipotez doğrulama aşamasında ise aday havaalanı bölgelerinde ÖBÖD özellikleri hesaplanmıştır ve bu özelliklerden FV üretilmiştir. FV betimleyicisinde görsel sözlük olarak Gauss Karışım Modeli (GKM) kullanılmıştır. GKM parametresinin logaritmik olasılığının

(log-likelihood) gradyanları, veri örneklerinin üretken sürecini modellemek için hesaplanmıştır ve tüm Gaussların kısmi türevleri birleştirilerek FV elde edilmiştir. FV, özellik olarak kullanılmıştır ve sınıflandırma adımında DVM'yi beslemiştir.



Şekil 1.1. Önerilen birinci havaalanı tespit yöntemine ait akış şeması

Önerilen yaklaşım, aşağıdaki katkıları sağlamıştır:

1-) Literatürde DPA yöntemine ek olarak DÇ haritası da ilave edilerek aday havaalanı bölgelerinin tespit edilme başarımının artırıldığı gözlenmiştir. Ancak bu tez çalışmasında DÇ haritası kullanılmamıştır. Geliştirdiğimiz DPA yöntemi ile aday havaalanı bölgesi çıkarımı sürecinde harcanan zaman, doğrulukta herhangi bir azalma olmaksızın azaltılmıştır. Ayrıca kullanılan veri setinin tamamında tespit edilen aday bölgelerde, gerçek havaalanı bölgesinin de tespit edildiği gözlenmiştir.

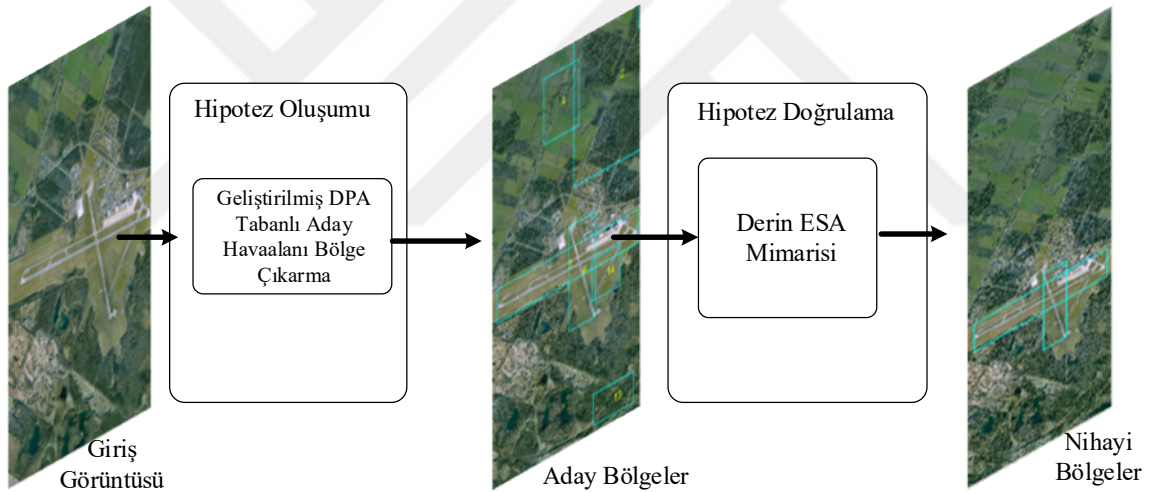
2-) ÖBÖD özelliklerini doğrudan kullanmak yerine önerdiğimiz yöntemde basit doğrusal sınıflandırmalarda bile iyi performans gösteren FV ile kodlanmış ÖBÖD özellikleri kullanılmıştır.

İkinci öneride ise Şekil 1.2’de blok şeması verilen yöntem sunulmuştur. İki aşamadan oluşan yöntem, birinci önerideki geliştirilmiş DPA ile birlikte ESA algoritmasını da içermektedir.

Birinci aşamada aday havaalanlarının tespiti yapılmaktadır. Algoritmada DPA ile başladıktan sonra elde edilen doğru parçalarından kısa olanlar elenmektedir. Ardından doğru parçaları arasında ilişkilendirme yapmak için etiketleme işlemi uygulanmaktadır. Düzenleme ve bağlama adımlarından sonra tekrar etiketleme işlemi gerçekleştirilmektedir.

DPA tabanlı algoritmanın son adımında ise çok kısa ve uzun doğru parçaları elenmektedir. Bu işlem sürecinin sonunda nihai aday hava alanları tespit edilmiş olmaktadır.

İkinci aşamada ise havaalanı tespit problemlerine karşı derin ESA mimarisi kullanılmıştır. Son zamanlarda derin ESA'lar, bilgisayarlı görme alanında çok sayıda uygulama ile göze çarpmaktadır. Genelde literatürde havaalanlarını tespit etmek için öncelikle çeşitli ayırt edici özelliklerin çıkarıldığı ve bir sınıflandırıcının kullanıldığı yaklaşım tercih edilmektedir. Ancak ESA'lar sadece bir özellik vektörü çıkarmanın yanı sıra daha iyi sınıflandırma doğruluğu sağlamaktadır. Bu nedenle bu tez çalışmasında havaalanı tespiti için ESA mimarisi de kullanılmıştır. Elde edilen UAG'lerin bir kısmı ESA mimarisini eğitmek için kullanılmıştır. ESA modeli, beş evrişim katmanı ve üç tam bağlı katmana sahiptir. Karşılaştırmalı çalışma ile önerilen yöntemin üstünlüğü doğrulanmış ve deneysel çalışmalar sonucunda oldukça iyi başarımlar elde edilmiştir.



Şekil 1.2. Önerilen ikinci havaalanı tespit yöntemine ait akış şeması

1.4. Tezin Organizasyonu

Tezin birinci bölümünde havaalanlarının önemi ve genel yapısı, literatürdeki benzer çalışmalar ve tezin organizasyonundan bahsedilmiştir.

Bölüm 2’de kenar ve doğru parçası tespit algoritmaları hakkında teorik bilgi verilmiştir.

Bölüm 3’te havaalanı tespit problemlerine karşı geliştirilen DPA, elde edilen havaalanı aday görüntülerinden dokusal özelliklerin çıkarımı için ÖBÖD ve FV kodlama, sınıflandırıcı için kullanılan DVM’den bahsedilmiştir. Ayrıca bu bölümde önerilen ikinci yöntemi

değerlendirmek için kullanılan hem özellik çıkarma hem de sınıflandırmayı bir arada barındıran derin ESA mimarisi de detaylıca açıklanmıştır.

Bölüm 4’te önerilen yöntemlere ilişkin parametreler, literatürdeki yöntemler ile başarımlar ve süre bazında kıyaslama yapılmıştır.

Bölüm 5’te tezin sonuçları irdelenmiş ve tez çalışmasına yapılan orijinal katkılar vurgulanmıştır.



2. KENAR VE DOĞRU PARÇASI ALGILAYICILARI

Kenar tespiti algoritmaları görüntü işleme ve bilgisayarlı görme uygulamalarında yaygın bir şekilde kullanılmaktadır. Kenar tespitinin temel amacı, bir görüntüde keskin süreksizlikleri bulmak ve tanımlamaktır. Bu süreksizlikler, nesnelerin sınırlarını karakterize eden piksel yoğunluğundaki ani değişikliklerden kaynaklanmaktadır. Kenarlar görüntüdeki farklı bölgeler arasındaki sınırları ifade etmektedir. Bu sınırlar nesne tanıma uygulamalarında bölütleme ve eşleştirme için kullanılmaktadır [22]. Bu nesne sınırları, kenar tabanlı yüz tanıma, engel tespiti ve hedef tanıma gibi birçok bilgisayarlı görme uygulamalarında ilk adımdır. Bu sebeple kenar algılayıcıları kenar tespit uygulamaları için gerekli hale gelmiştir.

Kenar tespiti için çeşitli operatör türleri mevcuttur. Bu operatörler iki kategoride incelenmektedir. Birinci dereceden türevde [23] kenarların eşiklenerek tespit edildiği gradyan görüntülerin üretilmesi için girdi görüntüsü adaptif bir maske ile evriştirilmektedir. Sobel, Prewitt ve Roberts gibi en klasik kenar algılayıcıları birinci dereceden türev operatörleridir [24]. Bu operatörlere gradyan operatörler de denilmektedir. Bu gradyan operatörler kenarları maksimum ve minimum yoğunluk değerlerine bakarak tespit ederler. Başka bir ifade ile verilen bir pikselin komşuluğunda yoğunluk değerlerinin dağılımlarını araştırır ve o pikselin bir kenar olarak sınıflandırılıp sınıflandırılmayacağını belirler. Ancak bu operatörler daha fazla hesaplama süresine sahip olduğundan gerçek zamanlı uygulamalarda kullanılamazlar. İkinci dereceden türevde operatör [23], görüntüdeki maksimumların varlığını belirleyen sıfır geçiş noktalarının çıkarılmasına dayanmaktadır. Öncelikle görüntü adaptif bir filtre ile yumuşatılır [25]. İkinci dereceden türev gürültüye karşı çok hassas olduğu için filtreleme işlemi oldukça önemlidir. Bu operatörler Marr ve Hildreth tarafından önerilen bir Gauss-Laplace'ından (LoG) türemektedirler [25]. Burada görüntü bir Gauss filtresi ile yumuşatılmaktadır. Bu operatör için Gauss filtresinin varyansı ve eşik değerler gibi bazı parametrelerin düzeltilmesi gerekmektedir. Otomatik hesaplama için bazı yöntemler de mevcuttur [26]. Ancak çoğu durumlarda bu parametre değerleri kullanıcı tarafından ayarlanmalıdır. LoG'un önemli bir problemi, filtrelemenin yumuşatma etkisi ile sıfır geçiş noktalarındaki artan sapmalardan kaynaklanan kenarların yerinin doğru tespit edilememesidir [27]. Bu problem için çözüm ise Canny tarafından önerilmiştir [28]. İyi tespit, yerelleştirme ve sadece tek bir kenar elde etme gibi özelliklere sahip olmasıyla

birlikte adım kenar tespiti için ideal bir operatördür. Daha sonra ise diğer operatörler önerilmiştir [29-32]. Bu operatörler gürültülü görüntülere karşı iyi verimlilik sağlamaktadır. Ancak olması gereken yerlerin dışında farklı kenar türleri tespit ettiğinden dolayı kenar yerelleştirme konusunda başarısız yönleri mevcuttur [33].

2.1. Roberts, Sobel ve Prewitt Kenar Algılayıcıları

Sobel [34], Roberts [35], Prewitt [36] gibi gradyan tabanlı algılayıcılar bir gradyan görüntüsü üretmek için görüntü girdisini kendi evrişim maskeleri ile evriştirmektedirler. Bu durum Tablo 2.1’de gösterilmiştir. Eşik değerleri kenarları saptamak için kullanılmaktadır. Kenar algılayıcıların çıkışı eşik değerine oldukça duyarlıdır. Bu operatörlerin Matlab uygulaması adaptif bir δ eşiği kullanmaktadır. Bu eşikler görüntüdeki ortalama karekök tahminine bağlıdır.

Tablo 2.1. Roberts, Sobel ve Prewitt evrişim maskeleri

Kenar Tespit Operatörü	Evrişim Maskesi G_x	Evrişim Maskesi G_y
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Gradyan genliği aşağıdaki denklem ile ifade edilmektedir:

$$|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| |G_y| \quad (2.1)$$

Tablo 2.2. Kenar algılayıcılarının yönelim açıları

	Sobel ve Prewitt	$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$
Yönelim açısı	Roberts	$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) - 3\pi/4$

2.2. LoG Kenar Algılayıcısı

Bu kenar algılayıcısı kenarları yerelleştirmek için piksel yoğunluğunun ikinci dereceden türevini kullanmaktadır [37]. Bir $I(x,y)$ görüntü girdisinin $L(x,y)$ Laplace'ı aşağıdaki denklemde ifade edilmektedir ve Tablo 2.3'te gösterildiği gibi evrişim maskelerinin herhangi biri ile evriştirilerek hesaplanabilmektedir:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.2)$$

Bu operatör gürültüye duyarlıdır ve sıklıkla görüntüye Gauss yumuşatma filtresi ile yumuşatıldıktan sonra uygulanmaktadır. Gauss yumuşatma filtresi G_σ aşağıdaki denklem ile ifade edilmektedir:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.3)$$

Gauss yumuşatma, Tablo 2.3'e gösterildiği gibi farklı σ ölçeklerindeki maskelerden biri ile görüntüyü evriştirerek elde edilebilmektedir.

Tablo 2.3. Farklı ölçeklerde LoG maske yapıları

Kenar tespit operatörü	Evrişim maskeleri	
LoG	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

LoG operatörü, sıfır geçiş noktalarını ve ikinci dereceden türev haritalarında görüntü gradyan genliklerini birleştirerek kenarları tespit etmektedir [38,39]. Kenarın alt piksel konumunu belirlemek için doğrusal tahmini kullanmaktadır. LoG kenar algılayıcısının davranışı büyük ölçüde LoG filtresinde kullanılan Gauss yumuşatma filtresinin standart sapması tarafından yönetilmektedir. Daha büyük σ değeri için Gauss filtresi daha geniştir ve daha fazla yumuşatma yapılmış olmaktadır. Ancak çok fazla yumuşatma yapılması kenar tespitini bir hayli zorlaştırmaktadır. Çünkü kenarlar, yüksek yoğunluklu gradyan noktaları olarak düşünülmektedir.

2.3. Canny Kenar Algılayıcısı

Bu algılayıcıda kenar tespitini geliştirmek için bazı ölçütler ele alınmaktadır. Birincisi gerçek kenarların kolaylıkla bulunabilmesinden dolayı düşük hata oranına sahiptir. İkincisi iyi yerleştirilmiş kenar noktalarının olmasıdır. Diğer bir deyişle kenar piksellerin konumunun kenar sınır bölgeleri üzerinde olmasıdır. Üçüncüsü ise tek bir kenara sadece tek bir yanıt vermesidir. Canny kenar algılayıcısının uygulanabilmesi için aşağıdaki adımlar takip edilmelidir:

Adım 1: Öncelikle kenarlar tespit edilmeden önce giriş görüntüsünden gürültüler filtrelenebilir. Bu işlem için Gauss filtresi kullanılmaktadır. Gauss yumuşatması evriştirme yöntemi kullanılarak yapılabilmektedir. Bir defada pikseli kare olarak işleyen ve görüntü üzerinde kaydırılan maske boyutu düşük olmalıdır. Maskenin genişliği dikkatlice seçilmelidir ve bu genişlik yerleştirme hatasıyla doğru orantılıdır.

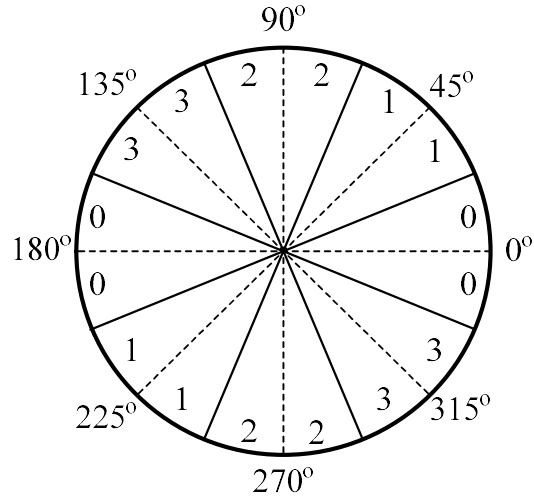
Adım 2: Kenar kuvveti, görüntünün gradyanının elde edilmesiyle bulunmaktadır. Bu amaç için Roberts ya da Sobel maskesi kullanılabilir [34,35]. Gradyan genişliği aşağıdaki denklem yardımıyla bulunmaktadır:

$$|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \quad (2.4)$$

Adım 3: x ve y yönlerindeki gradyanları kullanarak kenar yönleri bulunmaktadır. Bu da aşağıdaki denklem ile bulunmaktadır:

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.5)$$

Adım 4: Kenar yönü belirlendikten sonra belirli açı ile ilişkilendirilir. Kenar yönü yatay, dikey ya da diyagonal olarak çözümlenmektedir [2]. Şekil 2.1’de kenarların hangi açılar ile ilişkilendirileceğine dair dilimler gösterilmiştir.

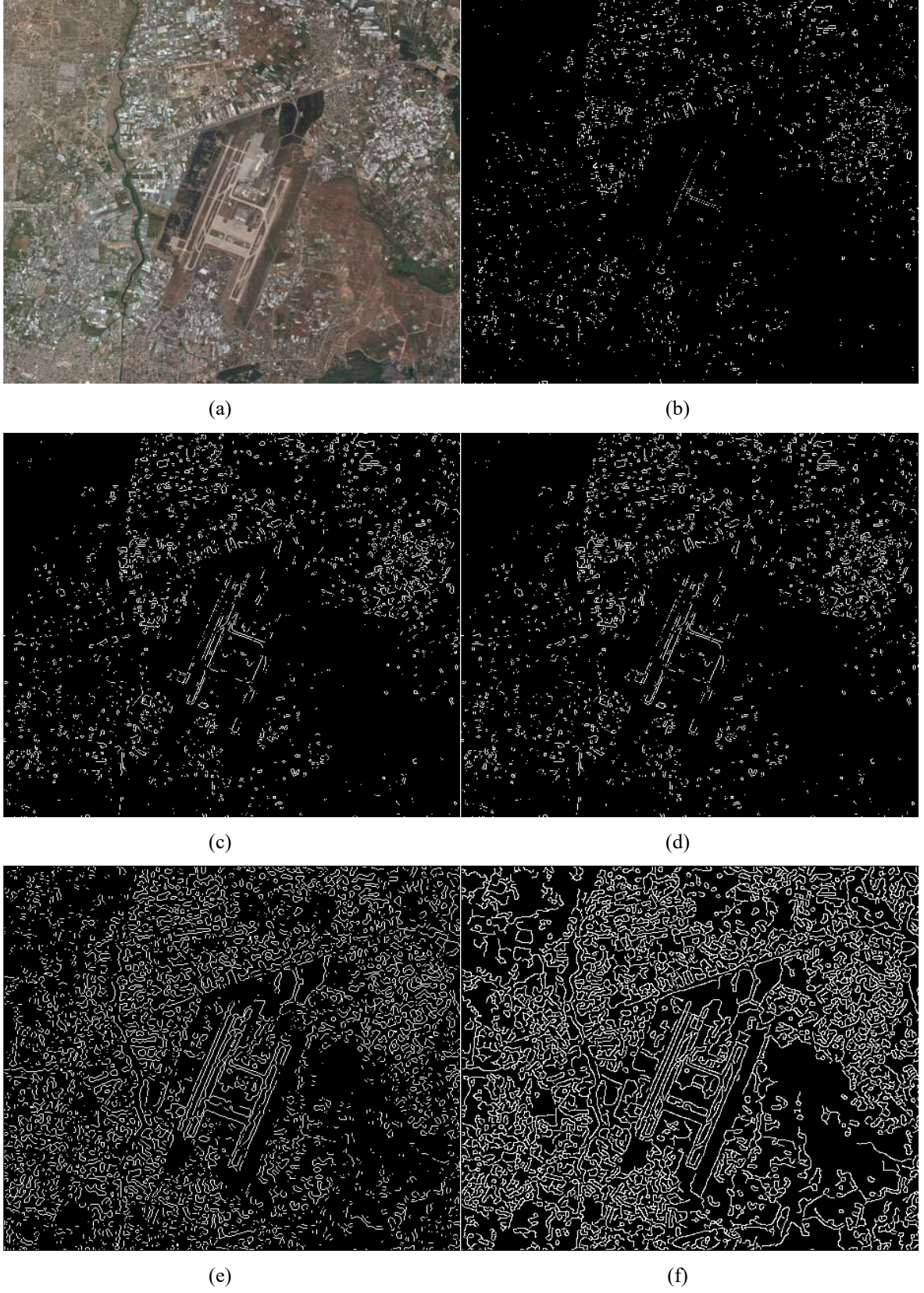


Şekil 2.1. Kenarların belirli açılarla ilişkilendirilmesi

Adım 5: Bu aşamada maksimum olmayanlar baskılanmaktadır. Başka bir şekilde ifade edilirse kenarlar üzerinde kenar olarak kabul edilmeyen bazı pikseller bulunmaktadır. Bu pikseller sıfır olarak ayarlanmaktadır. Bu sayede çıkış görüntüsünde ince kenar çizgileri oluşturulmaktadır.

Adım 6: Son olarak iz oluşumunu elimine etmek için 5. adımdaki çıkış görüntüsüne iki adet eşik değeri uygulanmaktadır [40].

Yukarıda bahsedilen kenar bulma algoritmalarına ait sonuçlar Şekil 2.2’de gösterilmiştir.



Şekil 2.2. Kenar bulma sonuçları; (a) Orijinal görüntü, (b) Roberts operatörü, (c) Sobel operatörü, (d) Prewitt operatörü, (e) LoG operatörü, (f) Canny operatörü

2.4. Doğru Parçası Algılayıcısı

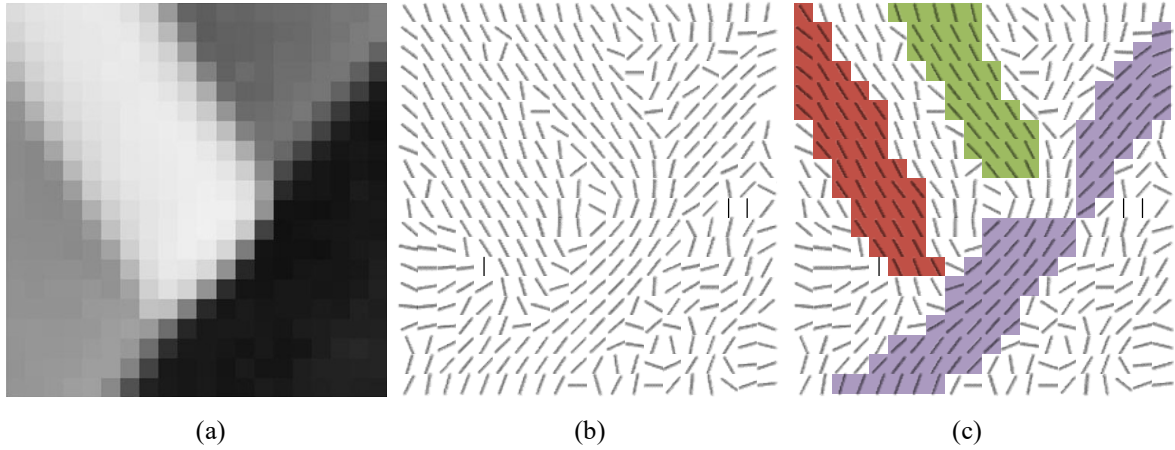
DPA algoritması, 2008 yılında R. Von Gioi ve arkadaşları tarafından önerilmiştir [11]. Algoritma 3 temel adımdan oluşmaktadır:

Adım 1: Doğru destek bölgelerindeki sıralı olarak komşuluğu olan grup pikseller, önceden belirlenmiş bir tolerans açısına kadar olan gradyan yönelimlerini benzer olarak kabul etmektedir.

Adım 2: Doğru destek bölgelerine en yakın bir doğru parçası bulunmaktadır.

Adım 3: Contrario model tabanlı yaklaşım ile doğru parçalarını doğrulanmaktadır.

Adım 1 ve 2, temel fikirlerini kaynak [41]'den almıştır ve bu doğrultuda bazı iyileştirmeler yapılmıştır. Her iki pikseli bir doğru parçası olarak bağlamaya çalışmak yerine [41], Adım 1'de bir bölge büyütme algoritması kullanılmıştır. Öncelikle daha büyük gradyan genlikli bir piksel asıl nokta olarak seçilir ve bölge açısı, piksellerin gradyan açısına dik olan seviye-doğru açısına ayarlanır. Daha sonra bölge büyütme algoritması başlar. Doğru destek bölgesine yakın olan her piksel test edilir. Eğer bir piksel, belirli bir tolerans açısına kadar tüm bölge ile benzer seviye doğru açısına sahip ise o piksel doğru destek bölgesine eklenecektir. Şekil 2.3'te doğru destek bölgesinin elde edilmesi gösterilmiştir.



Şekil 2.3. (a) Orijinal gri seviyeli görüntü, (b) seviye-doğru alanı, (c) doğru destek bölgesi

Doğru destek bölgesine her yeni bir piksel eklendiğinde, bölge açısı aşağıdaki gibi güncellenmektedir:

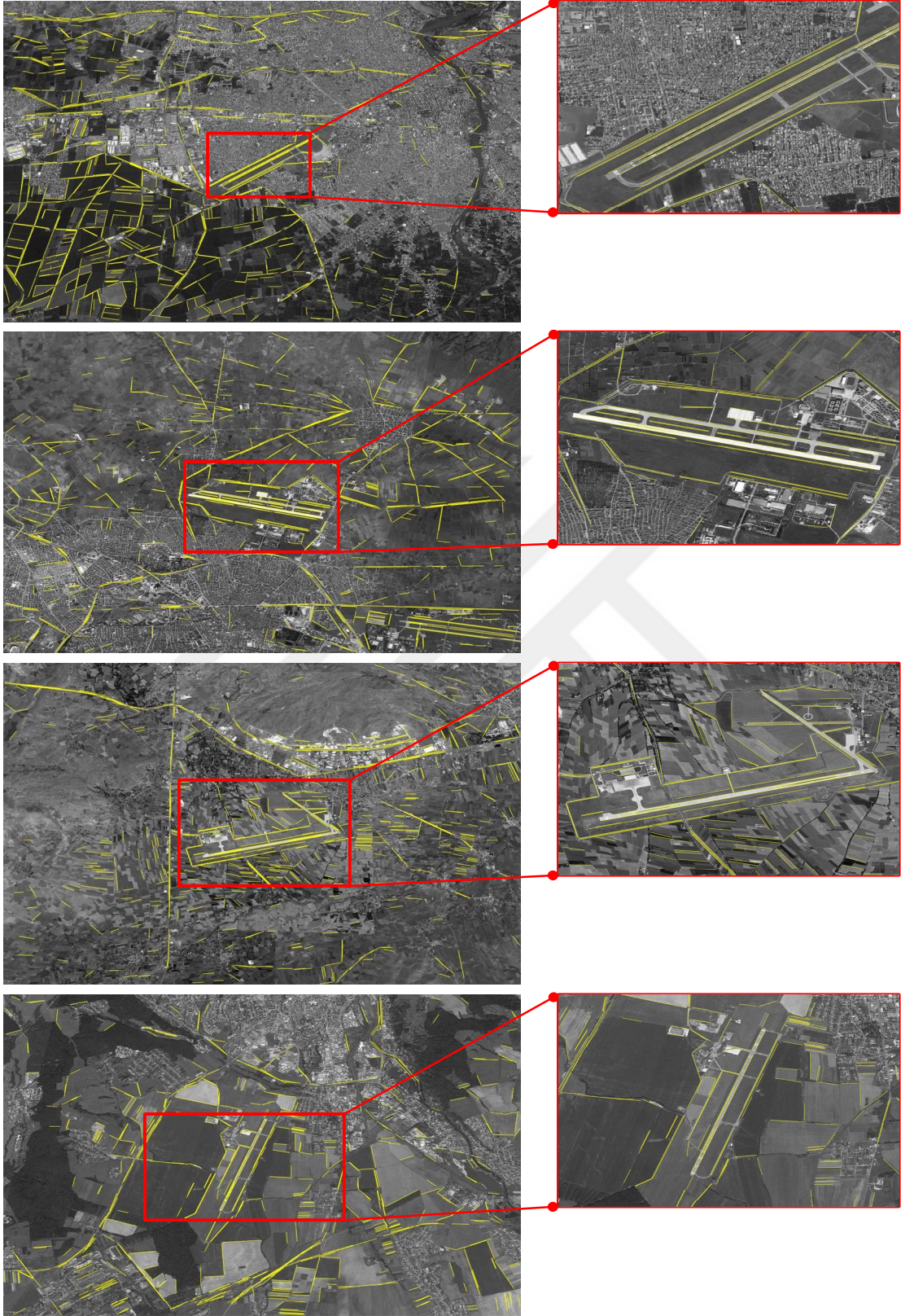
$$bölge_{açı} = \arctan \frac{\sum \sin(açı)}{\sum \cos(açı)} \quad (2.6)$$

Burada ‘açı’ iki parça arasındaki açıdır. Bu işlem, mevcut olan doğru destek bölgesine eklenecek yeni bir piksel kalmayana kadar tekrarlanmaktadır. Ayrıca her pikselin sadece bir doğru destek bölgesine ait olduğu unutulmamalıdır ve böylece bu işlem doğrusal çalışma süresine sahip olmaktadır.

Doğru destek bölgeleri tamamen oluşturulduktan sonra bir dikdörtgen ile temsil edilen doğru parçaları belirlenmeye çalışılmaktadır. Bilindiği gibi bir dikdörtgenin uzunluğu, genişliği, merkezi ve yönelim açısı mevcuttur. Doğru parçası algılayıcısına gelince doğru destek bölgesinin merkezi, kendi kütle merkezi ve temel atalet eksenini ise dikdörtgenin yönelim açısı olarak seçilmektedir.

Son adım doğru parçasının doğru olarak tanımlanıp tanımlanmadığına karar vermektir. Doğru parçası algılayıcısı, kaynak [42] ve [43]’deki basitleştirilmiş varsayım test problemi olarak tespiti gerektiren bir *Contrario* yaklaşımını kullanmaktadır.

Kaynak [41]’de yazarlar 22,5 derecelik bir tolerans açısının en iyi sonucu verdiğini iddia etmiş ve binlerce görüntü ile bu iddiayı doğrulamışlardır. Şekil 2.4’te uydu görüntülerinin gri seviyesine DPA yönteminin uygulama sonucu gösterilmiştir.



Şekil 2.4. Doğru parçası algılayıcısının sonucu

3. ÖNERİLEN HAVAALANI TESPİT YÖNTEMİ

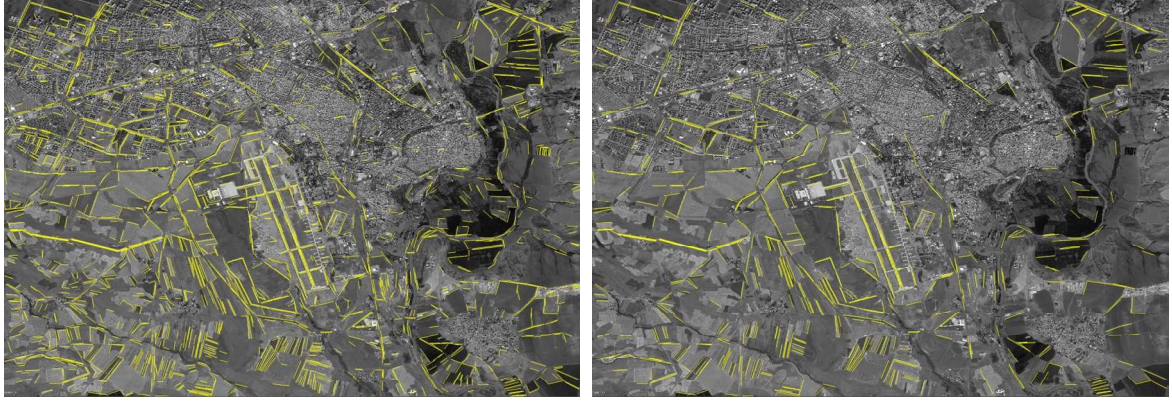
3.1. Havaalanı Aday Bölgelerin Tespiti İçin Geliştirilmiş Doğru Parçası Algılayıcısı

Havaalanı tespit algoritmalarında temel yaklaşım, havaalanı aday bölgelerin bulunması esasına dayanmaktadır. Aday bölgeler bulunurken genellikle kenar ya da doğru tespiti yapılmaktadır. Bu doğrultuda literatürde kullanılan bazı kenar algılayıcılarından Sobel, Roberts, Prewit, Canny ve LoG algoritmaları bir önceki bölümde kısaca açıklanmıştı. Ancak Şekil 2.2’de elde edilen sonuçlar incelendiğinde havaalanının yapısal özelliklerine ters düşen kısa ve eğri çizgilerin de elde edildiği gözlenmiştir. Havaalanı pistinin temel özelliklerinden biri de uzun ve düz bir yapıya sahip olmasıdır. Bu sebeple aday bölgelerin tespiti için yine bir önceki bölümde bahsedilen DPA algoritması seçilmiştir.

DPA, gürbüz bir doğru parçası algılayıcısı olarak 2008 yılında Gioi ve arkadaşları tarafından önerilmiştir [11]. Algoritma, her pikseldeki seviye-doğru açısını hesaplayarak başlamaktadır. Daha sonra belirli bir toleransa kadar aynı seviye-doğru açısına sahip piksellerin bölgeleri bağlanılarak bir seviye-doğru alanı oluşturmaktadır. Bağlanmış alanlar, doğru destek bölgeleri olarak adlandırılmaktadır. Son olarak Contrario model tabanlı doğru parçalarının doğrulama eylemi yerine getirilmektedir.

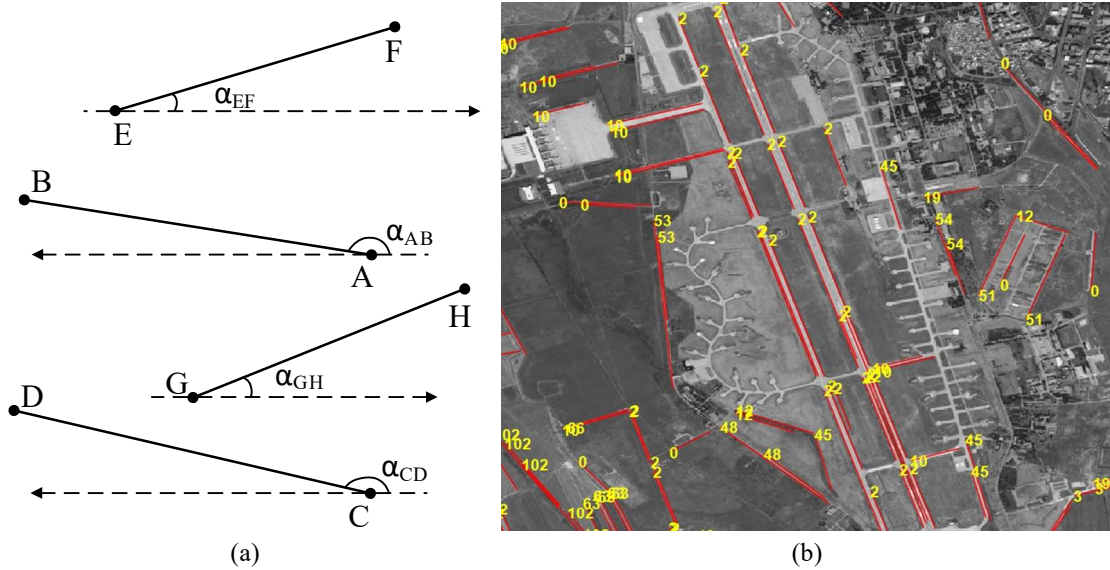
DPA, dayanıklı ve geniş parametre ayarı imkanı sunan bir algoritmadır. Ancak en büyük dezavantajı bağlanmamış doğru parçalarına sahip olmasıdır. Bu bağlanmamış doğru parçaları, havaalanı pistinin tespitini zorlaştırmaktadır. Bu dezavantaj ile başa çıkmak adına havaalanı pistinin tespiti için yeni bir algoritma geliştirdik. Geliştirdiğimiz yeni algoritmanın aday havaalanı çıkarma adımları aşağıdaki gibidir:

Adım 1: Öncelikle birçok doğru parçasının üretildiği UAG girdisine DPA algoritması uygulanmaktadır. Önceden tanımlanmış thr_{dl} eşik değerinden küçük uzunluklara sahip doğru parçaları Şekil 3.1’de gösterildiği gibi elimine edilmektedir.



Şekil 3.1. Doğru parçalarının eliminasyonu, (a) DPA'nın çıkışı, (b) Eliminasyon işlemi sonrası

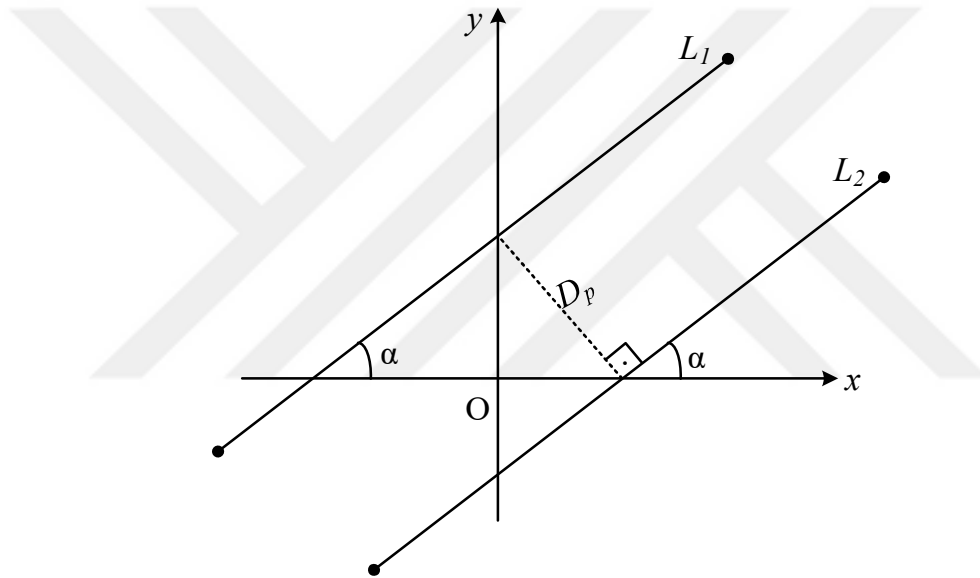
Adım 2: Bu adımda X yatay eksenine göre α açısı, her doğru parçası için hesaplanmaktadır. Bu açılara göre doğru parçalarının kümelenmesi için etiketleme işlemi gerçekleştirilmektedir. Eğer iki doğru parçasının açı farkı önceden tanımlı bir thr_θ eşik açısından küçük ve bu doğru parçaları birbirlerine yeteri kadar yakın ise (iki doğru parçası arasındaki mesafe thr_{d2} eşik değerinden küçük olma durumu) etiket numaraları aynı olmaktadır. Bu işlem Şekil 3.2(a)'da gösterilmektedir. Örneğin; X eksenine açıları arasındaki fark $|\alpha_{AB} - \alpha_{CD}| < thr_\theta$ eşitliğini sağlıyorsa $|AB|$ ve $|CD|$ doğru parçaları aynı etiket ile işaretlenirler. Benzer bir şekilde $|\alpha_{EF} - \alpha_{GH}| < thr_\theta$ eşitliği sağlanırsa $|EF|$ ve $|GH|$ doğru parçaları da aynı etiketle gruplandırılırlar.



Şekil 3.2. (a) Doğru parçası etiketleme işleminin gösterimi, (b) Havaalanı görüntüsünde doğru parçası etiketleme

Bir havaalanı görüntüsünde bu adımın uygulanması Şekil 3.2(b)'de gösterilmiştir. Şekilden de görülebileceği üzere, hemen hemen paralel ve birbirlerine yakın olan doğru parçaları aynı etiket ile işaretlenmiştir. Bu kuralları sağlamayan doğru parçaları ise sıfır “0” etiketi ile işaretlenmektedir. Sonuç olarak sıfır ile etiketlenmiş doğru parçaları elimine edilmiş olmaktadır.

Adım 3: Bu adım bir düzenleme adımı olarak görülmektedir. Düzenleme için L_1 ve L_2 paralel doğru parçası arasındaki dik mesafenin hesaplanması gerekmektedir. Denklem (3.1)'de Şekil 3.3'e ait D_p dik mesafenin hesaplaması gösterilmektedir:



Şekil 3.3. İki paralel L_1 ve L_2 doğru parçaları ve onlar arasındaki D_p dik mesafe

$$\begin{aligned}
 L_1 : ax + by + c_1 &= 0 \\
 L_2 : ax + by + c_2 &= 0 \\
 D_p &= \frac{|c_1 - c_2|}{\sqrt{a^2 + b^2}}
 \end{aligned} \tag{3.1}$$

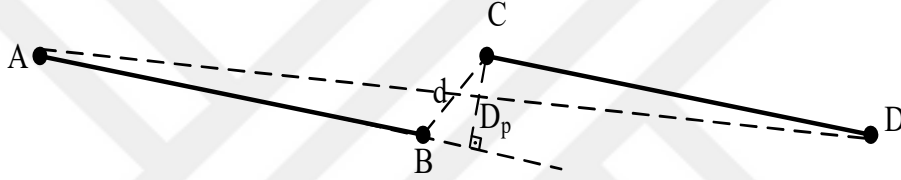
Adım 2'de α açısı thr_θ eşik açısından daha küçük ise bu iki doğru parçası hemen hemen paralel olarak kabul edildiğinden, Denklem (3.1) doğrudan uygulanamamaktadır. Bu problemi çözmek için öncelikle doğru parçalarının α açıları en uzun doğru parçasının α açısına göre düzenlenmektedir ve daha sonra her bir doğru parçasının başlangıç ve bitiş noktaları buna bağlı olarak tekrar hesaplanmaktadır.

Adım 4: Bu adım bağlantı sürecini içermektedir. Doğru parçalarının bağlantısı her küme için ayrı ayrı elde edilmektedir.

Bir küme içerisinde aşağıda verilen bağlantı kriterini sağlayan bir çift doğru parçası, bağlanabilir çift olarak adlandırılmaktadır:

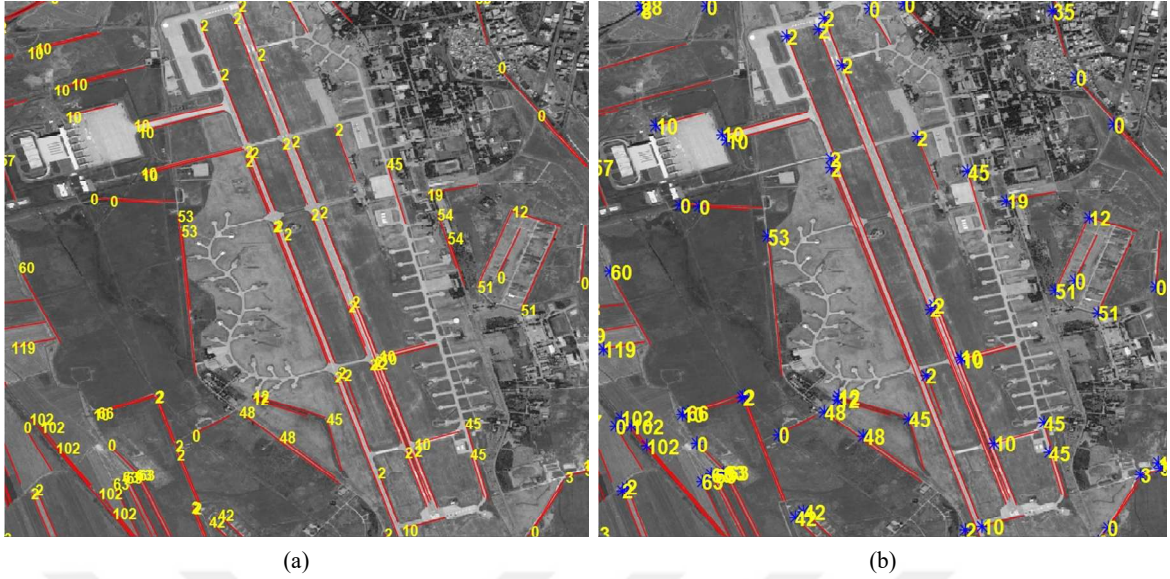
- Doğru parçaları çifti arasındaki D_p dik mesafe (Denklem (3.1) kullanılarak hesaplanmış), $D_p < thr_{d3}$ şartını sağlamış olmalı,
- Çift için bir doğru parçasının sonu ile diğer doğru parçasının başlangıç noktası arasındaki d mesafesi, $d < thr_{d4}$ eşitliğini sağlamış olmalı.

İki doğru parçasının bağlanabilir bir çift teşkil etmesi durumunda, bu çift kümeden çıkarılır ve tek bir doğru olarak bağlanıp küme içerisine yerleştirilir. Bağlanabilir doğru parçası çifti, kümede bağlanabilir çift kalmayıncaya kadar tekrarlanarak bağlanırlar.



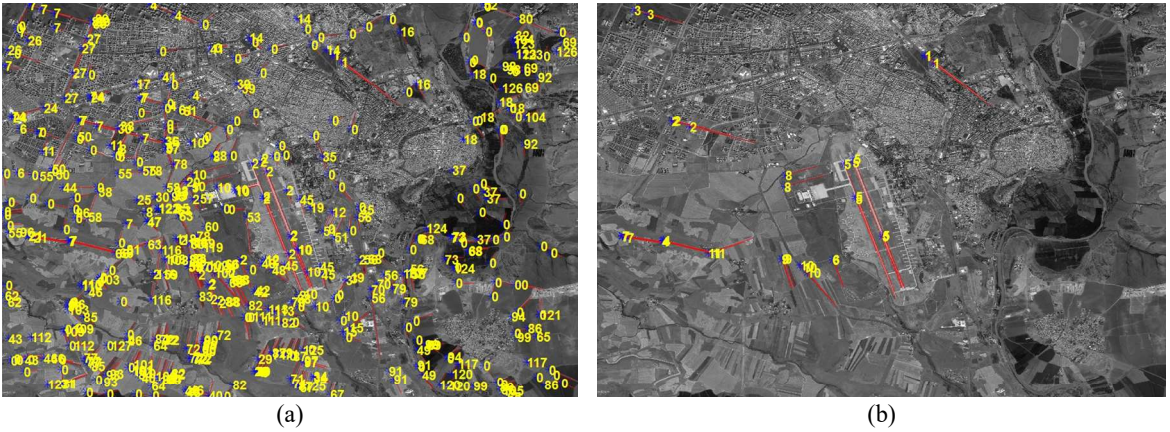
Şekil 3.4. İki doğru parçasının bağlantısı

Bağlantı sürecini daha iyi anlamak için bir örnek Şekil 3.4'te gösterilmektedir. Adım 2'de $|AB|$ ve $|CD|$ doğru parçalarının aynı etiket ile işaretlendiğini varsayalım. $|AB|$ ile $|CD|$ arasındaki D_p dik mesafesi thr_{d3} eşik değerinden küçük ve B ile C noktaları arasındaki d mesafesi başka bir thr_{d4} eşik değerinden küçük ise $|AB|$ ile $|CD|$ doğru parçaları atılır ve yerine yeni bir $|AD|$ doğru parçası üretilir. Bu bağlantı işlemi tekrarlanan bir süreçtir ve tüm bağlantılar tamamlandığında işlem durmaktadır. Bir UAG'de başka bir bağlantı örneği Şekil 3.5'te verilmiştir. Buradan da görüleceği üzere havaalanı pisti 2 sayısı ile etiketlenmiştir ve pistin karşıt kenarlarında 2 ve 3 adet doğru parçaları mevcuttur. Diğer etiketlenmiş doğru parçaları da 2 sayısı ile etiketlenmiştir. Bağlantı süreci uygulandıktan sonra havaalanı pistinin sadece iki doğru parçası kalmaktadır. Dahası diğer aynı etikete sahip doğru parçaları, bağlantı kriterlerini sağlamadığı için piste bağlanmamaktadır.

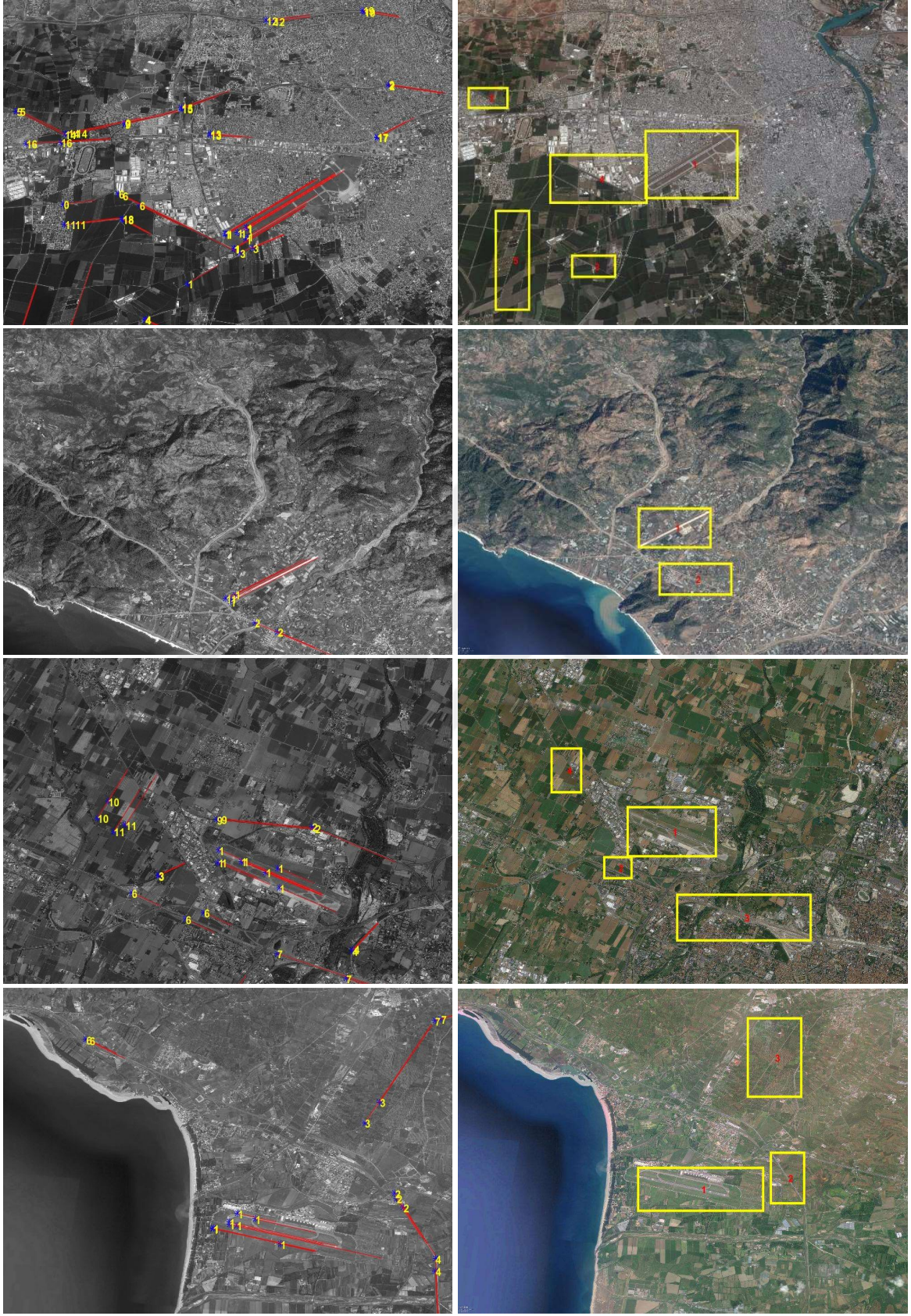


Şekil 3.5. UAG’de bağlantı prosedür sonucu, (a) Adım 4’ten önce, (b) Adım 4’ten sonra

Adım 5: Bağlantı sürecinden sonra doğru parçalarının uzunlukları ve α açıları tekrar hesaplanmaktadır. Daha sonra α açıları kullanılarak bağlanmış doğru parçalarından yeni bir doğru parçası etiketleme işlemi elde edilmektedir. Adım 2’de hem doğru parçalarının α açı farkları hem de konumlarına dayanarak bu etiketleme süreci elde edilmişti. Adım 2’ye benzer bir şekilde 0 etiketli doğru parçaları elimine edilmektedir. Adım 3’te verilen düzenleme süreci tekrar uygulanmaktadır. Daha sonra doğru parçası eliminasyon süreci her kümede uzak doğru parçalarının elenmesi için uygulanmaktadır. Bir kümede bir doğru parçasının en uzun doğru parçasına olan paralel uzaklığı verilen bir $thras$ eşik değerinden büyük ise bir doğru parçası çıkarılabilmektedir. Böylece potansiyel havaalanı bölgelerini içeren etiketlenmiş doğru parçaları Şekil 3.6’da gösterilmiştir.



Şekil 3.6. Doğru parçası azaltma, (a) Adım 5’ten önce, (b) Adım 5’ten sonra



(a)

(b)

Şekil 3.7. Nihai doğru parçası eliminasyonu (a) Adım 6'dan önce, (b) Adım 6'dan sonra

Adım 6: Bu adım nihai doğru parçalarının eliminasyonunu içermektedir. Çok uzun ya da çok kısa doğru parçaları çıkarılmaktadır. Çünkü belirli çözünürlüklü bir UAG'de pistin uzunlukları sınırlı bir aralıkta değişmektedir. Geriye kalan doğru parçalarının aday havaalanı bölgeleri olduğu varsayılır ve bir dikdörtgen içerisine alınmaktadır. Tespit edilen nihai havaalanı aday bölgeleri Şekil 3.7'de gösterilmiştir.

3.2. Ölçekten Bağımsız Özellik Dönüşümü

Bu yöntem ilk olarak 2004 yılında David G. Lowe tarafından önerilmiştir [44]. Aşağıda, görüntü özellik kümesini elde etmek için kullanılan hesaplamaların en temel aşamaları verilmiştir:

1. *Ölçek-uzay yerel maksimum ve minimumların (kilit noktalar) tespiti:* Havaalanı aday bölgelere ait görüntüden farklı ölçeklerde ya da boyutta görüntüler oluşturulur. Farklı boyuttaki her görüntü setine oktav denir. Her oktavda farklı σ (sigma) değerli Gauss filtresi ile yumuşatılan görüntülere Gauss farkı fonksiyonu uygulanır. Bu işlem ile görüntünün farklı ölçeklerde kenarları vurgulanır. Gauss farkı ile elde edilen kenarlardan maksimum ve minimum değere sahip olan bölgeler adreslenir. Böylece ölçekten ya boyuttan bağımsız bir yapı oluşmaktadır.
2. *Doğru Kilit noktaların lokalizasyonu:* Aday kilit noktalar belirlendikten sonra her bir kilit nokta için lokasyon ve ölçek bilgisi belirlenir. Ardından düşük karışıklık (kontrast) değerine sahip olan ya da kenar üzerinde konumlanmamış kilit noktalar çıkarılabilir.
3. *Yönelim atama:* Kilit noktaların genliği en yüksek olanların açısına göre görüntünün yönelim açısı tayin edilir. Böylece dönmeye karşı bağımsız bir yapı elde edilir.
4. *Kilit nokta tanımlama:* Bu aşamada ise görüntünün hangi aydınlık seviyesinde ve nesnenin hangi görüş açısından elde edildiğinin bir önemi kalmamaktadır.

ÖBÖB aşamaları detaylı olarak Bölüm 3.2'nin alt başlıklarında anlatılmıştır.

3.2.1. Ölçek-Uzay Yerel Maksimum ve Minimumların Tespiti

Bölüm 3.2'de anlatıldığı gibi aday bölgeleri tanımlamak için basamaklı filtreleme yaklaşımı kullanılarak kilit noktalar tespit edilmektedir. Kilit nokta tespitinin ilk aşaması,

aynı nesnenin farklı görünüşleri altında tekrarlanabilir şekilde atanabilecek yerleri ve ölçekleri tanımlamaktır. Görüntünün ölçek değişimlerine bağımsız olan kararlı özellikleri aramak suretiyle bölgelerin tespiti, ölçek alanı olarak bilinen sürekli bir ölçek fonksiyonu kullanılarak gerçekleştirilebilmektedir [45].

Farklı makul varsayımlar altında tek mümkün olan ölçek-uzay çekirdeğinin Gauss fonksiyonu olduğu Koenderink [46] ve Lindeberg [47] tarafından açıklanmıştır. Böylece bir görüntünün ölçek uzayı, değişken ölçekli $G(x, y, \sigma)$ evrişiminin $I(x, y)$ görüntüsünden üretilen bir $L(x, y, \sigma)$ fonksiyonu olarak tanımlanmaktadır:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.2)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (3.3)$$

burada $*$, evrişim operatörüdür.

Kaynak [48]'de ölçek uzayda kararlı kilit nokta konumlarını etkili bir şekilde tespit etmek için görüntü ile evrişime tabi Gauss fonksiyonunun $G(x, y, \sigma)$ farklarında sabit bir k çarpım faktörü ile iki yakın ölçek farkından hesaplanabilen ölçek-uzay uç noktalarının kullanılması önerilmiştir:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.4)$$

Bu fonksiyonu seçmenin birçok nedeni mevcuttur. İlk olarak özellikle yumuşatılmış L görüntüsü, ölçek-uzay özellik çıkarma için etkili bir fonksiyondur. Böylece D fonksiyonu, basit bir görüntü farkı ile elde edilebilmektedir. Bununla birlikte Kaynak [47]'de incelendiği gibi Gauss fonksiyon farkları, ölçeği normalize edilmiş $\sigma^2 \nabla^2 G$ Laplace Gauss fonksiyonuna yakın bir yaklaşım sağlamaktadır. Lindeberg, σ^2 faktörlü Laplace normalizasyonunun doğru bir ölçek bağımsızlığı için gerekli olduğunu göstermiştir [47]. Ayrıntılı deneysel karşılaştırmalarda $\sigma^2 \nabla^2 G$ 'nin maksimum ve minimumları, diğer fonksiyonlara kıyasla en kararlı görüntü özelliklerini ürettiğini bulmuştur [49]. D ve $\sigma^2 \nabla^2 G$ arasındaki ilişki, ısı dağılım eşitliği ile anlaşılmaktadır (σ^2 yerine σ parametresi ile ifade edilmiştir):

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (3.5)$$

Buradan $\nabla^2 G$ 'nin $k\sigma$ ve σ yakın ölçeklerin farkları kullanılarak $\partial G/\partial\sigma$ sonlu farklar yaklaşımından hesaplanabildiği görülmektedir:

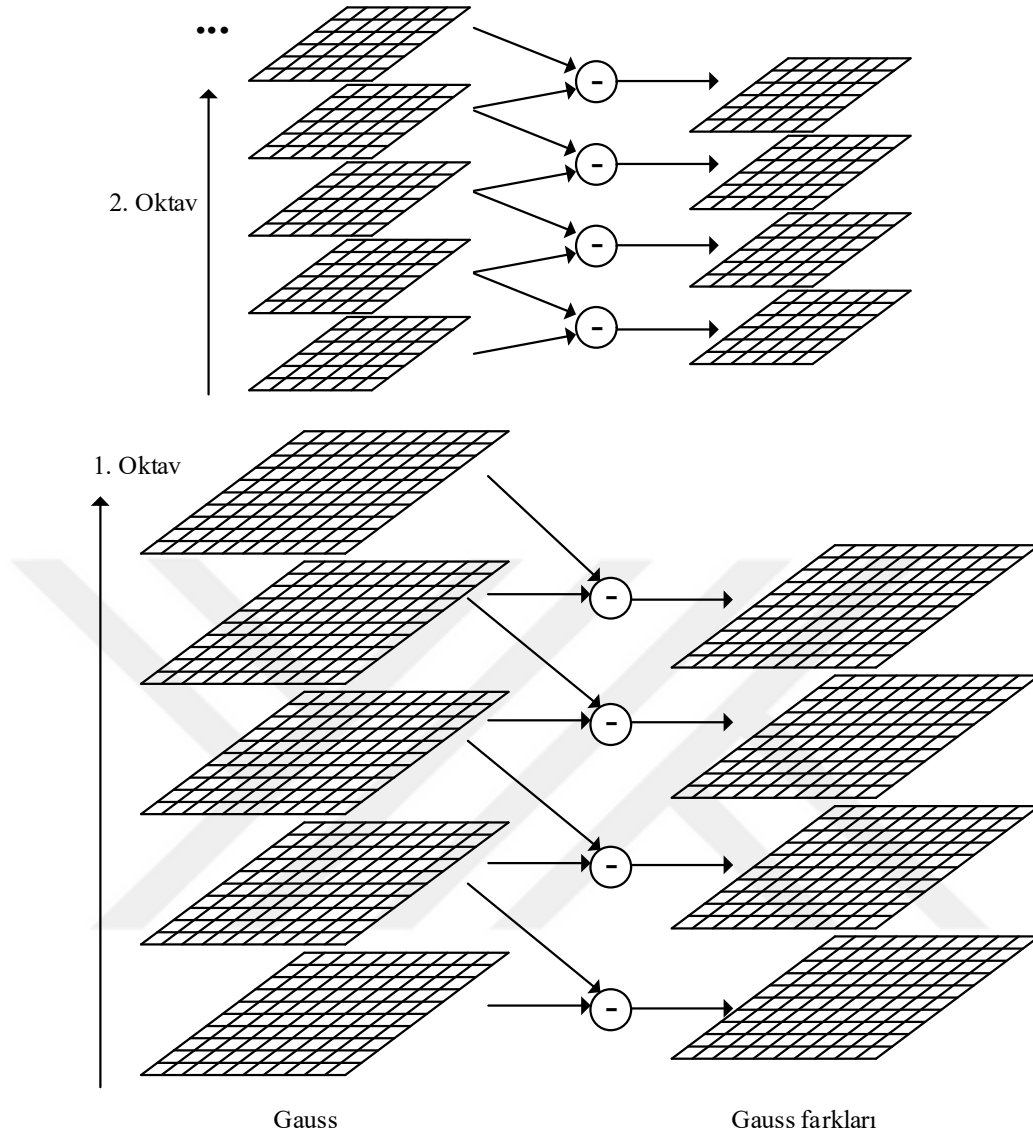
$$\frac{\partial G}{\partial\sigma} = \sigma \nabla^2 G \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (3.6)$$

Böylece,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G \quad (3.7)$$

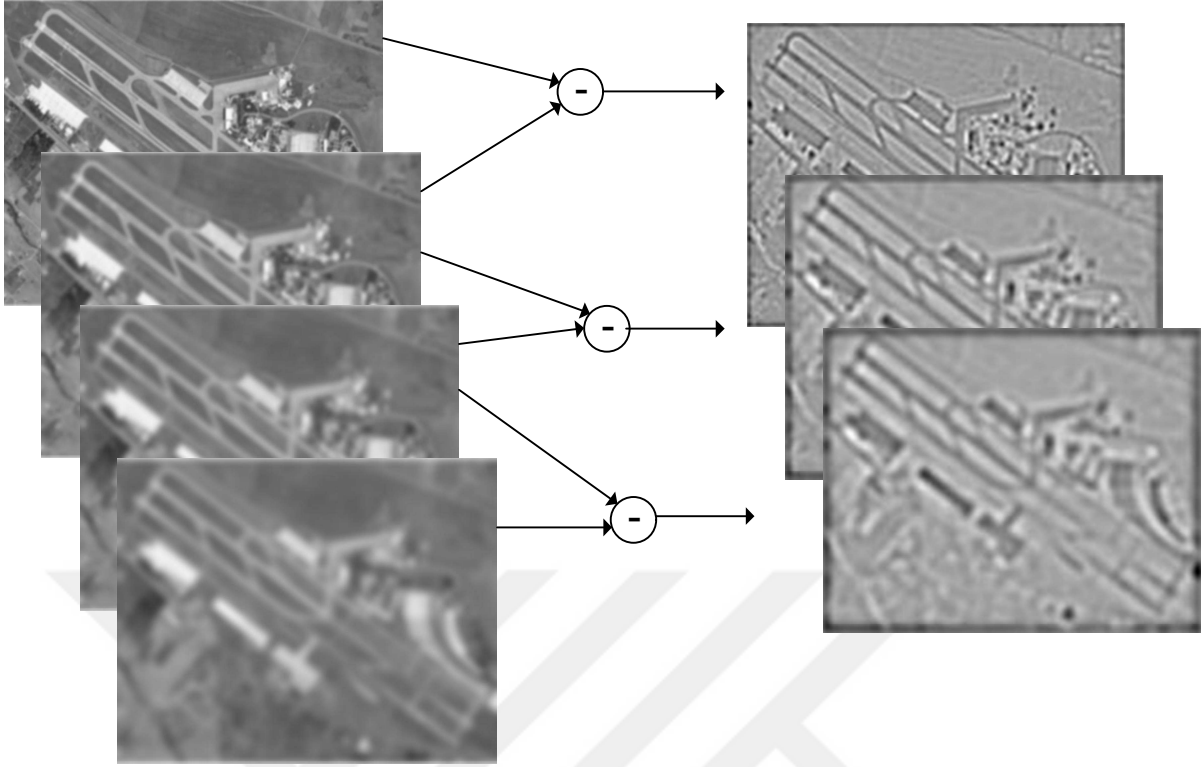
olur. Bu da Gauss fark fonksiyonunun sabit bir faktör ile farklı ölçeklere sahip olduğunda, ölçekten bağımsız Laplace için gerekli olan σ^2 ölçeğinin zaten eklenmiş olduğu görülmektedir. Eşitlikteki $(k-1)$ faktörü tüm ölçeklerde sabittir ve bu nedenle maksimum ve minimum konumlar etkilenmemektedir. k , 1'e doğru yaklaştığında yaklaşık hata sıfıra doğru gidecektir. Ancak pratikte $k = \sqrt{2}$ gibi bir ölçek katsayısında yaklaşık değerin, maksimum ve minimumların stabilizesinde hemen hemen hiç etki etmediği görülmüştür.

Bir $D(x, y, \sigma)$ yaklaşımının yapısı Şekil 3.8'de gösterilmiştir. Sol kolonda görüldüğü üzere başlangıç görüntüsü ölçek uzayda sabit bir k faktörlü Gauss görüntüsünü üretmek için kademeli olarak evriştirilmektedir. Aynı ölçekli ancak farklı σ değeri ile yumuşatılmış bulanık görüntüler tam bir oktavı oluşturmaktadır. Sağ kolondaki Gauss fark görüntülerini üretmek için de birbirine en yakın ölçekli görüntülerin farkı alınır. Bir tam oktav işlendiğinde iki adet yumuşatılmış görüntünün farkları ile yeni görüntüler elde edilmiş olmaktadır. Farklı k faktörü için diğer oktavlarda yapılan işlemler daha düşük ölçeklerde olması sebebiyle az işlem yükü gerektirse de, ilk oktavda yapılan işlem basamaklarıyla aynı yol izlenmektedir.



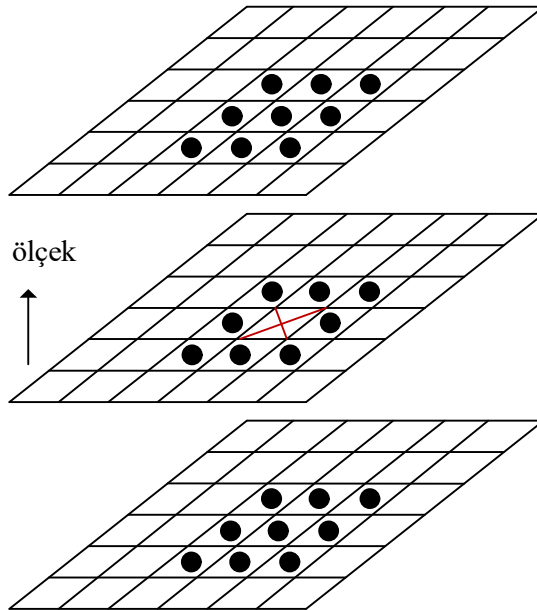
Şekil 3.8. Ölçek uzayının her oktavında farklı Gauss ölçeği ile yumuşatılmış görüntü setinin elde edilmesi (sol sütun), ve bitişik Gauss görüntülerinin farkı (sağ sütun)

Şekil 3.8'deki durumun daha iyi anlaşılabilmesi için Gauss fark fonksiyonunun, geliştirdiğimiz DPA ile elde ettiğimiz havaalanı aday bölgesine uygulanmış hali Şekil 3.9'da gösterilmiştir.



Şekil 3.9. Bir oktav için aday bölgelerden Gauss farklarının elde edilmesi

$D(x, y, \sigma)$ Gauss farklarının yerel maksimum ve minimumlarının tespiti için örnek nokta, hem kendi komşuluğundaki 8 nokta hem de alt ve üst ölçeğe karşılık gelen 9'ar komşuluk ile karşılaştırılır. Bu durum Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Yerel maksimum ve minimumların tespit edilmesi

Örnek nokta sadece karşılaştırılan diğer 26 noktadan büyük ya da küçük ise yerel maksimum ya da minimum olarak seçilmektedir. Yerel maksimum ve minimum ismi de buradan gelmektedir. Çoğu örnek noktası ilk birkaç kontrolün ardından elimine edileceğinden hesaplama maliyeti oldukça düşük olacaktır.

Önemli bir konu, görüntüdeki örnekleme frekansını ve yerel maksimum ve minimumların güvenilir bir şekilde tespiti için gerekli ölçek alanının belirlenmesidir. Ancak yerel maksimum ve minimumlar keyfi olarak birbirine yakın olabileceğinden belirli bir ölçek alanı yoktur.

3.2.2. Doğru Kilit Noktaların Lokalizasyonu

Bir örnek nokta ile komşuluklarını karşılaştırarak kilit nokta bulunduğundan sonra bir sonraki adım; konum, ölçek ve temel eğriliklerin oranı için kilit noktanın etrafında uyumlu bir bilgi sağlamaktır. Bu bilgi düşük kontrasta sahip noktaların elenmesi prensibine dayanmaktadır.

Bu yaklaşımın ilk uygulaması Lowe tarafından gerçekleştirilmiştir [48]. Ancak yakın zamanda Brown ara değer olarak enterpole edilmiş maksimumların yerlerini belirlemek için yerel örnek noktalara 3B karesel fonksiyonunun kullanıldığı bir yöntem geliştirmiştir [50] ve deneyleri eşleşme ve kararlılık için önemli bir gelişme sağladığını göstermektedir. Bu yaklaşım $D(x, y, \sigma)$ ölçek-uzay fonksiyonunun Taylor genişlemesini kullanmaktadır. Başlangıç noktası örnek noktasına gelecek şekilde kaydırılmıştır:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (3.8)$$

Burada D ve türevleri örnek noktada değerlendirilir ve $x = (x, y, \sigma)^T$ ise bu nokta dengedir (offset). Maksimum ve minimumların yerleri $\hat{x}$, bu fonksiyonunun x 'e göre türevi alınıp sıfıra eşitleyerek belirlenir:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (3.9)$$

Brown tarafından önerildiği gibi Hessian matrisi ve D 'nin türevi komşu örnek noktaların farkları kullanılarak hesaplanmaktadır. Bu da 3×3 'lük doğrusal bir sistem ile minimum maliyetle çözülebilir. Eğer $\hat{x}$ herhangi bir ölçekte 0,5'ten büyük ise yerel maksimum ve minimumların konumları farklı bir örnek noktasına daha yakındır anlamına gelmektedir. Bu durumda örnek noktası değiştirilir ve enterpolasyon gerçekleştirilir. Son $\hat{x}$ ofseti yerel maksimum ve minimumların yerleri için örnek noktanın konumuna eklenir.

Yerel maksimum ve minimumlardaki $D(\hat{x})$ fonksiyonun değerleri düşük kontrasta sahip noktaları elimine etmek için kullanışlıdır. Bu durum denklem (3.9)'u, denklem (3.8)'e ekleyerek elde edilmektedir:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (3.10)$$

Lowe'a göre $D(\hat{x})$ 'nin 0,03'ten küçük olan tüm değerler elimine edilmiştir (görüntünün 0 ile 1 aralığında olduğu varsayılır) [44].

Kararlılık için düşük kontrasta sahip kilit noktaların elimine edilmesi yeterli olmamaktadır. Her ne kadar kenarların etrafında kötü bir şekilde tespit edilen kilit noktalar olsa da Gauss fonksiyonunun farkı kenarlarda güçlü bir etki göstermektedir. Böylece kararsız küçük miktarlarda gürültü mevcut olmaktadır. Bu gürültüleri ortadan kaldırmak için de kilit noktalarının önceden tanımlanmış olan yer ve ölçeğine 2×2 'lik bir H Hessian matrisi uygulanmaktadır:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (3.11)$$

Türevler komşu örnek noktalarının farkları dikkate alınarak tahmin edilmektedir. H 'nin özdeğerleri, D 'nin temel eğrilikleri ile orantılıdır. Harris and Stephens tarafından kullanılan bu yaklaşımla [51], özdeğerlerin açık bir şekilde hesaplanması önlenebilmektedir. α en büyük genlikli ve β ise daha küçük bir özdeğer olsun. Hessian matrisinden özdeğerlerin toplamı ve onların çarpımları aşağıdaki denklemlerle hesaplanabilmektedir:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.12)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.13)$$

Hesaplama sonunda muhtemel negatif değerler olması durumunda, ilgili bölge gürültü kabul edildiğinden yani yerel maksimum ve minimum noktaları olmadığından atılırlar.

En büyük özdeğerli α ile ondan daha küçük değere sahip β 'nin birbirine oranı $\alpha = \beta * r$ olsun;

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (3.14)$$

olur. Bu durumda bu değerler sadece kendi değerlerinden ziyade özdeğerlerin oranına da bağlı olmaktadır. İki özdeğer birbirine eşit olduğunda $(r + 1)^2/r$ minimum düzeyde olmaktadır ve r değeri ile artmaktadır. Böylece temel eğriliklerin oranını kontrol etmek için aşağıdaki eşitlik kullanılmaktadır:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r} \quad (3.15)$$

Bu eşitlik her kilit noktası için uygulanmaktadır. Bu çalışma için r değeri 10 olarak kullanılmıştır. Eşitliğin sağlanmadığı kilit noktalar elimine edilmektedir. Bu sayede doğru kilit noktaları tespit edilmiş olmaktadır.

3.2.3. Yönelim Atama

Her kilit noktasına, yerel görüntü özelliklerine dayalı tutarlı bir yönelim atayarak, bu kilit noktalar bir yönelime göre temsil edilmesi sağlanır ve böylece görüntünün yönü ne durumda olursa olsun dönmeden bağımsızlık özelliği kazandırılmış olmaktadır. Bu yaklaşım Schmid ve Mohr'un görüntünün dönme bakımından değişmez bir ölçüye dayandığı çalışmasıyla ispatlanmıştır [52].

Yerel bir yönelim tayini için en istikrarlı sonuç aşağıdaki denklem ile açıklanmıştır. Kilit noktasının ölçeği, yumuşatılmış en yakın ölçekli L Gauss görüntüsünü seçmek için kullanılmıştır. Böylece tüm hesaplamalar ölçekten bağımsız bir şekilde yapılmaktadır. Her örnek $L(x, y)$ görüntüsü için piksel farkları kullanılarak bu ölçekte $m(x, y)$ gradyan genliği ve $\theta(x, y)$ yönelim açısı hesaplanabilmektedir:

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2} \quad (3.16)$$

$$\theta(x, y) = \tan^{-1}[L(x, y+1) - L(x, y-1)] / [L(x+1, y) - L(x-1, y)] \quad (3.17)$$

Bir yönelim histogramı bir kilit noktasının etrafındaki bir bölgedeki örnek noktalarının gradyan yönelimlerinden oluşmaktadır. Yön histogramlarında 360 derecelik yönelim açılarını kapsayan 36 aralık bulunmaktadır. Histograma eklenen her örnek, gradyan büyüklüğüne ve σ ölçeğinin 1,5 katı olan dairesel bir Gauss penceresine göre ağırlıklandırılırlar.

Yön histogramlarındaki tepeler yerel gradyanların baskın yönlerine karşılık gelmektedir. Histogramdaki en yüksek tepe tespit edilir ve daha sonra bu tepenin %80'ine kadar olan tepeler de birer kilit nokta oluşturur. Böylece benzer büyüklükteki birden fazla tepe noktasına sahip yerler için aynı lokasyonda ve ölçekte ancak farklı yönelim açılarına sahip çoklu kilit noktalar elde edilmektedir. Şekil 3.11'de örnek bir havaalanı aday bölge üzerinde elde edilmiş kilit noktalar ve gradyan yönelimleri gösterilmiştir.

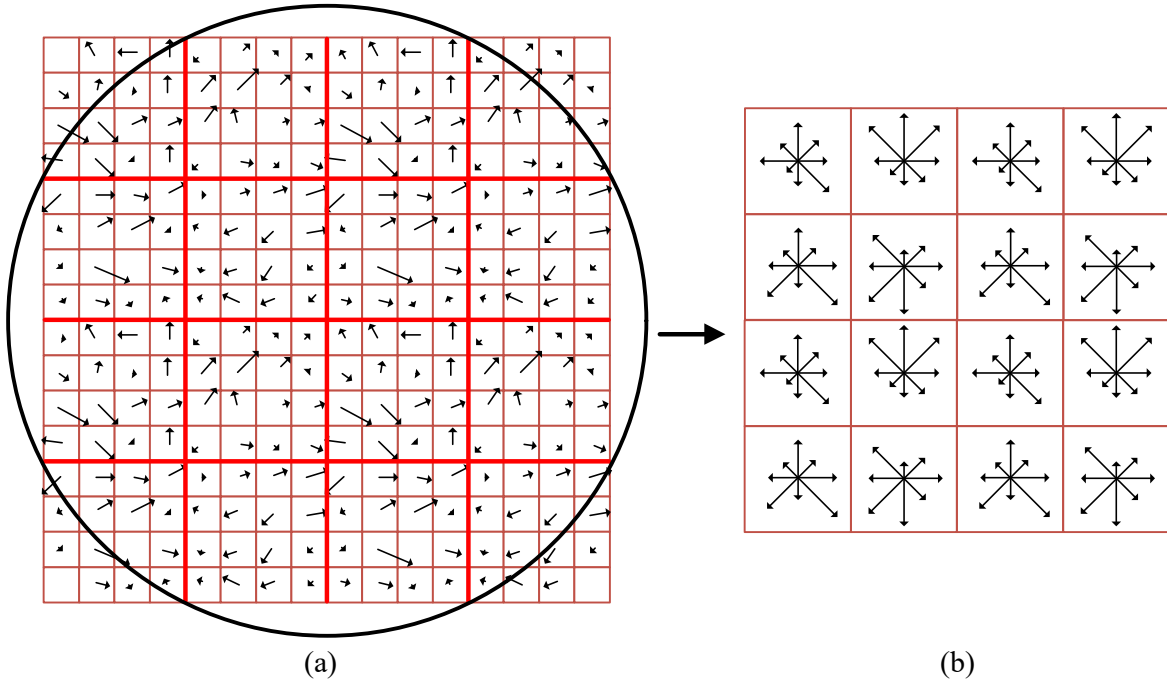


Şekil 3.11. Elimine işlemlerinden sonra havaalanı aday bölge üzerinde bulunan kilit noktalar

3.2.4. Kilit Nokta Tanımlama

Daha önceki bölümlerde aday bölgelere ait her kilit noktasına bir konum, ölçek ve yönelim açısı atanmıştı. Bu parametreler yerel görüntü bölgesini tanımlamak için tekrarlanabilir bir 2B koordinat sistemini kullanmaktadır ve bu da dolayısıyla parametrelere değişmezlik sağlamaktadır. Bu bölümde ise görüntünün bakış açısı ve ortamın aydınlık değişimlerine duyarsızlık özellikleri hesaplanacaktır. Açıkça yaklaşım, kilit nokta etrafındaki yerel görüntü yoğunluğunu uygun bir ölçekte örneklemek ve bunları normalleştirilmiş bir bağıntı ölçütü ile eşleştirmek olacaktır (Şekil 3.13 ve Şekil 3.14).

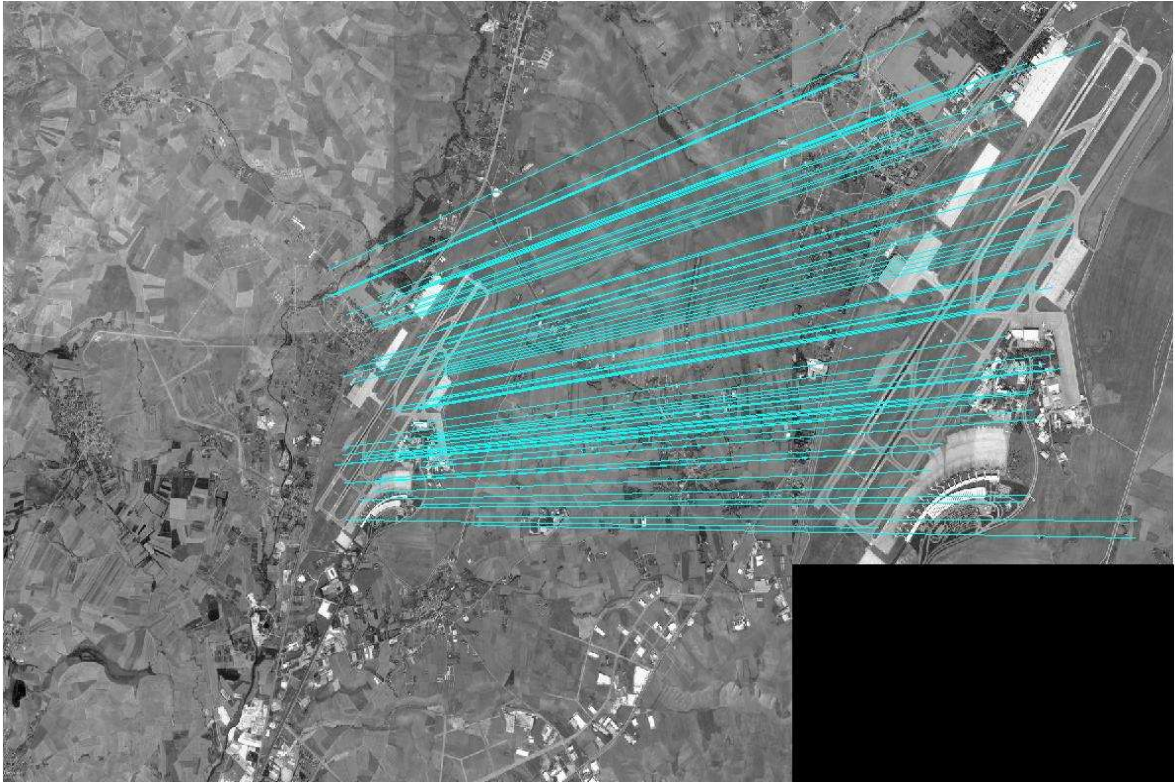
Öncelikle Gauss yumuşatmasının seviyesini seçmek için kilit noktanın ölçeği kullanılarak görüntü gradyan genlikleri ve yönelim açıları kilit nokta konumunun etrafında örneklenir. Yönelim değişmezliğini sağlamak için tanımlayıcının koordinatları ve gradyan yönelimleri, anahtar noktanın yönelim açısına göre döndürülür. Bölüm 3.2.3'te açıklandığı gibi iyi bir verim için gradyanlar, tüm oktav seviyesi için hesaplanmıştı. Bu durum Şekil 3.12(a)'da her örnek konumunda küçük oklarla gösterilmiştir. Her örnek noktasının büyüklüğüne bir ağırlık atamak için tanımlayıcı penceresinin genişliğinin yarısına eşit bir Gauss ağırlıklandırma fonksiyonu kullanılmaktadır.



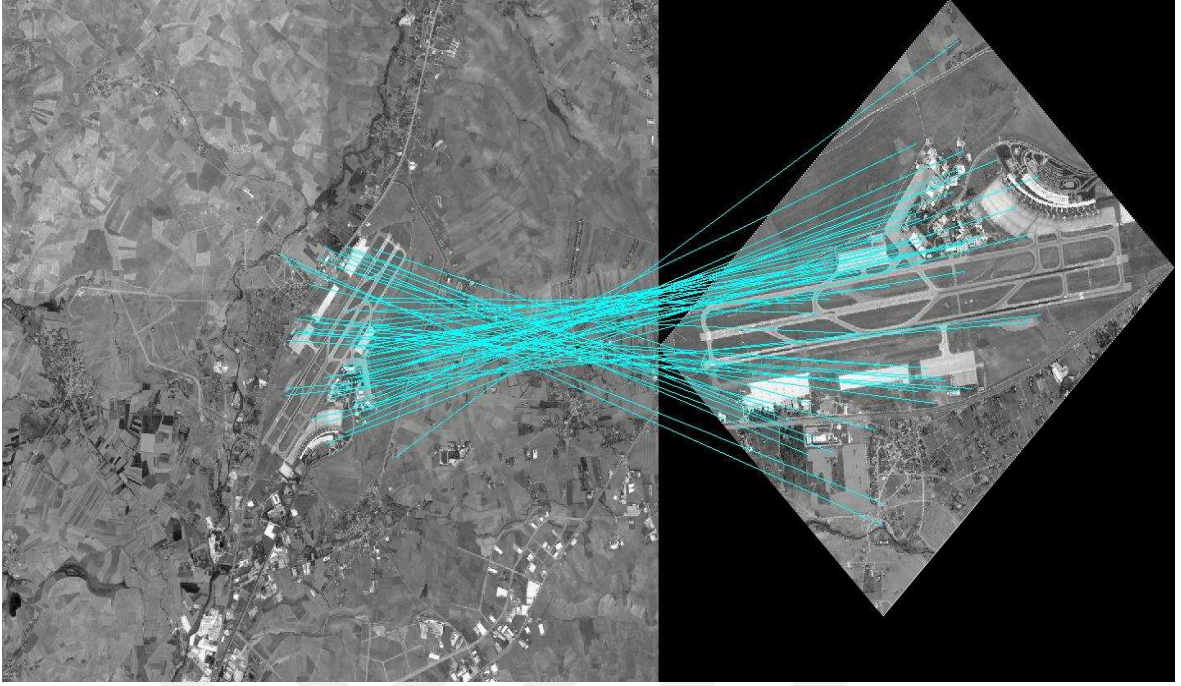
Şekil 3.12. (a) Görüntünün gradyanları, (b) anahtar nokta tanımlayıcı

Kilit nokta tanımlayıcısı Şekil 3.12(b)'de gösterilmektedir. İlk olarak kilit nokta etrafında 16×16 'lık komşu bölge alınmaktadır ve 4×4 boyutunda 16 adet alt blok oluşturulmaktadır. Her bir alt blok için 8 aralıktaki yönelim histogramı, örneklerin büyüklük ve yönelim değerleri hesaplanarak oluşturulmaktadır. Her okun uzunluğu bu histogramın büyüklüğüne ve yönü ise yönelim açısına karşılık gelmektedir. $4 \times 4 = 16$ adet histogram ve her birinde 8 aralığın olmasından dolayı her kilit nokta için özellik vektöründe $4 \times 4 \times 8 = 128$ öge bulunmaktadır. Böylece her bir kilit nokta için 128 boyutlu özellik vektörü elde edilmiş olmaktadır ve özellikler FV'yi beslemektedir.

ÖBÖD yöntemi ile kilit noktaların, ölçeğe ve dönmeye karşı gürbüz bir yapıya sahip olduğunu gösteren eşleşme sonucu Şekil 3.13 ve Şekil 3.14'te gösterilmiştir.



Şekil 3.13. Farklı ölçekteki görüntülerin eşleştirme işlemi



Şekil 3.14. Farklı ölçek ve yöndeki görüntülerin eşleştirme işlemi

3.3. Fisher Vektör Kodlama

FV gösterimine, havaalanı aday bölgelerden ÖBÖD özniteliklerin çıkarılmasıyla başlamaktadır. FV kodlaması, büyük bir öznitelik vektör kümesini (çıkarılan ÖBÖD özellikleri) yüksek boyutlu vektör gösteriminde toplamaktadır. Bu işlem sırasında özelliklere parametrik üretken model olan Gauss Karışım Modeli (GKM) uyarlanarak ve ardından parametrelerine göre modelin logaritmik olasılık türevi kodlanarak yapılmaktadır [53]. Özellikler ve GKM merkezlerinin her biri arasındaki birinci ve ikinci dereceden farkların ortalamasını yakalayan bir gösterim elde edilmiş olmaktadır. O halde bir FV, farkların kümelenmesiyle elde edilmektedir [54].

$X = \{x_t, t = 1, 2, \dots, T\}$, bir görüntüden ÖBÖD yöntemi ile çıkarılmış D -boyutlu ($D=128$) T yerel tanımlayıcı dizisidir. X 'in oluşum süreci, λ parametrelili bir u_λ olasılık yoğunluk fonksiyonu ile modellenen varsayılmaktadır [55,56] ve X örnekleri, aşağıdaki gradyan vektörü denklemi ile tanımlanmaktadır [53]:

$$G_\lambda^x = \frac{1}{T} \nabla_\lambda \log u_\lambda(X) \quad (3.18)$$

Burada ∇_λ ; gradyan operatörünü ve $\log u_\lambda(X)$ ise λ parametrelili X 'in logaritmik olasılık fonksiyonudur. Logaritmik olasılığın gradyanı, parametrelerin üretim sürecine katkısını açıklamaktadır. Bu gradyan vektörün boyutsallığı sadece λ 'daki parametrelerin sayısına bağlıdır (T yamasına bağlı değildir). Gradyan üzerindeki doğal çekirdek aşağıdaki denklem ile ifade edilmektedir [53]:

$$K_{FK}(X, Y) = G_\lambda^{X^T} F_\lambda^{-1} G_\lambda^Y \quad (3.19)$$

Burada F_λ , u_λ olasılık yoğunluk fonksiyonunun bilgi matrisidir aşağıdaki ifade ile gösterilmektedir:

$$F_\lambda = E_{X \sim u_\lambda} [\nabla_\lambda \log u_\lambda(X) \log u_\lambda(X)^T] \quad (3.20)$$

$F_\lambda = L_\lambda^T L_\lambda$ ve $K(X, Y)$ eşitlikleri için Choleskey ayrıştırırmayı kullanarak normalize edilmiş g_λ^X vektörü aşağıdaki denklem gibi yazılmaktadır:

$$g_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \nabla_\lambda \log u_\lambda(X) \quad (3.21)$$

Burada g_λ^X , X 'in Fisher Vektörü olarak tanımlanmaktadır. g_λ^X fisher vektörünün boyutu, G_λ^X gradyan vektörünün boyutuna eşittir. K_{FK} 'yi çekirdek olarak kullanan doğrusal olmayan bir çekirdek makinesi ile g_λ^X 'yi özellik vektörü olarak kullanan doğrusal çekirdek makinesi eşdeğerdir. FV formülasyonun avantajı doğrusal sınıflandırmaların çok etkili bir şekilde öğrenilebilmesidir.

3.3.1. Örnek Bir FV Uygulaması

$X = \{x_t, t = 1, \dots, T\}$ olan D -boyutlu yerel tanımlayıcı olsun [44]. Örneklerin bağımsız olduğunu varsayarsak her örnek için FV eşitliği aşağıdaki gibi yazılmaktadır:

$$g_\lambda^X = \sum_{t=1}^T L_\lambda \nabla_\lambda \log u_\lambda(x_t) \quad (3.22)$$

Böylece bu bağımsızlık varsayımı altında g_λ^x fisher vektörü, her tanımlayıcı için normalize edilmiş $L_\lambda \nabla_\lambda \log u_\lambda(x_t)$ gradyan istatistiklerinin toplamı olmaktadır. Bu işlem:

$$x_t \rightarrow \varphi_{FK}(x_t) = L_\lambda \nabla_\lambda \log u_\lambda(x_t) \quad (3.23)$$

doğrusal sınıflandırmaya daha uygun olan daha yüksek boyutlu bir uzayda x_t yerel tanımlayıcıların yerleştirilmesi olarak anlaşılabilmektedir.

Daha sonra olasılık yoğunluk fonksiyonu u_λ , GKM ile modellendiği kabul edilmektedir [57]. Bir bilgisayarlı görme literatüründe herhangi bir görüntüdeki yerel tanımlayıcıların üretim sürecini modelleyen bir GKM, “*olasılıksal görsel sözlük*” olarak bahsedilmiştir [58,59]. GKM modelinin K bileşenli parametreleri aşağıdaki gibi ifade edilmektedir:

$$\lambda = \{w_k, \mu_k, \Sigma_k, k = 1, \dots, K\} \quad (3.24)$$

Burada w_k, μ_k ve Σ_k sırasıyla k Gaussunun karışım ağırlığı, ortalama vektörü ve kovaryans matrisidir. Olasılık yoğunluk fonksiyonu u_λ :

$$u_\lambda(x) = \sum_{k=1}^K w_k u_k(x) \quad (3.25)$$

olur. Burada u_k , k gaussunu ifade etmektedir ve aşağıdaki denklem ile gösterilmektedir:

$$u_k(x) = \frac{1}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\} \quad (3.26)$$

ve u_λ olasılık yoğunluk fonksiyonunun geçerli bir dağılım olduğundan emin olmak için:

$$\forall_k : w_k \geq 0 \text{ ve } \sum_{k=1}^K w_k = 1 \quad (3.27)$$

olmalıdır. Daha sonra standart bir varsayım olan diyagonal kovaryans matrisleri, σ_k^2 varyans vektörü ile gösterilmektedir. Maksimum olabilirlik kriterini optimize etmek için maksimum

olasılık yaklaşımı kullanılarak GKM parametreleri, yerel tanımlayıcıların büyük bir eğitim kümesinde tahmin edilmektedir.

Ağırlık parametreleri için Krapac ve arkadaşları tarafından önerilen soft-max formülü kullanılmaktadır ve aşağıdaki denklem ile gösterilmiştir [60]:

$$w_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^K \exp(\alpha_j)} \quad (3.28)$$

α_k kullanarak tekrar parametreleştirmek, eşitlik (3.27)'deki kısıtlamaları engellemeye zorlamaktadır. GKM modelinin $\lambda = \{\alpha_k, \mu_k, \Sigma_k, k = 1, \dots, K\}$ parametreleri ile ilgili tek bir x_t tanımlayıcısının gradyanları:

$$\nabla_{\alpha_k} \log u_{\lambda}(x_t) = \gamma_t(k) - w_k \quad (3.29)$$

$$\nabla_{\mu_k} \log u_{\lambda}(x_t) = \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k^2} \right) \quad (3.30)$$

$$\nabla_{\sigma_k} \log u_{\lambda}(x_t) = \gamma_t(k) \left[\frac{(x_t - \mu_k)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right] \quad (3.31)$$

burada $\gamma_t(k)$, sonrasal olabilirlik olarak ta bilinen k gaussu için x_t 'nin yumuşak atamasıdır:

$$\gamma_t(k) = \frac{w_k u_k(x_t)}{\sum_{j=1}^K w_j u_j(x_t)} \quad (3.32)$$

x_t tanımlayıcılarının boyutluluğu D ile ifade edilmektedir. k gaussunun μ_k ortalama vektör ile ilgili olan D -boyutlu $g_{\mu,k}^x$ FV gradyanları ve σ_k standart sapma ile ilgili olan $g_{\sigma,k}^x$ FV'leri aşağıdaki matematiksel türevler ile ifade edilmektedir:

$$g_{\mu,k}^x = \frac{1}{T \sqrt{w_k}} \sum_{t=1}^T \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k} \right) \quad (3.33)$$

$$g_{\sigma,k}^x = \frac{1}{T\sqrt{2w_k}} \sum_{t=1}^T \gamma_t(k) \left[\frac{(x_t - \mu_k)^2}{\sigma_k^2} - 1 \right] \quad (3.34)$$

Nihai g_{λ}^x FV'ü, $g_{\mu,k}^x$ ve $g_{\sigma,k}^x$ vektörlerinin $k = 1, \dots, K$ için birleşimidir ve sonuç olarak $2 \times K \times D$ boyutlu özellik vektörü elde edilmektedir.

3.4. Destek Vektör Makineleri

DVM iki ya da çok sınıflı veriye ait bir nokta kümesini ayıran ve en iyi iki boyutlu uzay için doğruyu, üç boyutlu uzay için düzlemi ve çok boyutlu uzay için hiper düzlemi bulmaya çalışan bir sınıflandırıcı olarak tanımlanabilir. $\{x_i, y_i\}$, $i = 1, \dots, N$ biçiminde ve N ile örneklenmiş bir veri çifti kabul edilsin. Burada $x_i \in R^d$ ve hedef $y_i \in \{-1, +1\}$ aralığındadır. Hiper düzlem ise aşağıdaki gibi tanımlanmaktadır:

$$w^T x + b = 0 \quad (3.35)$$

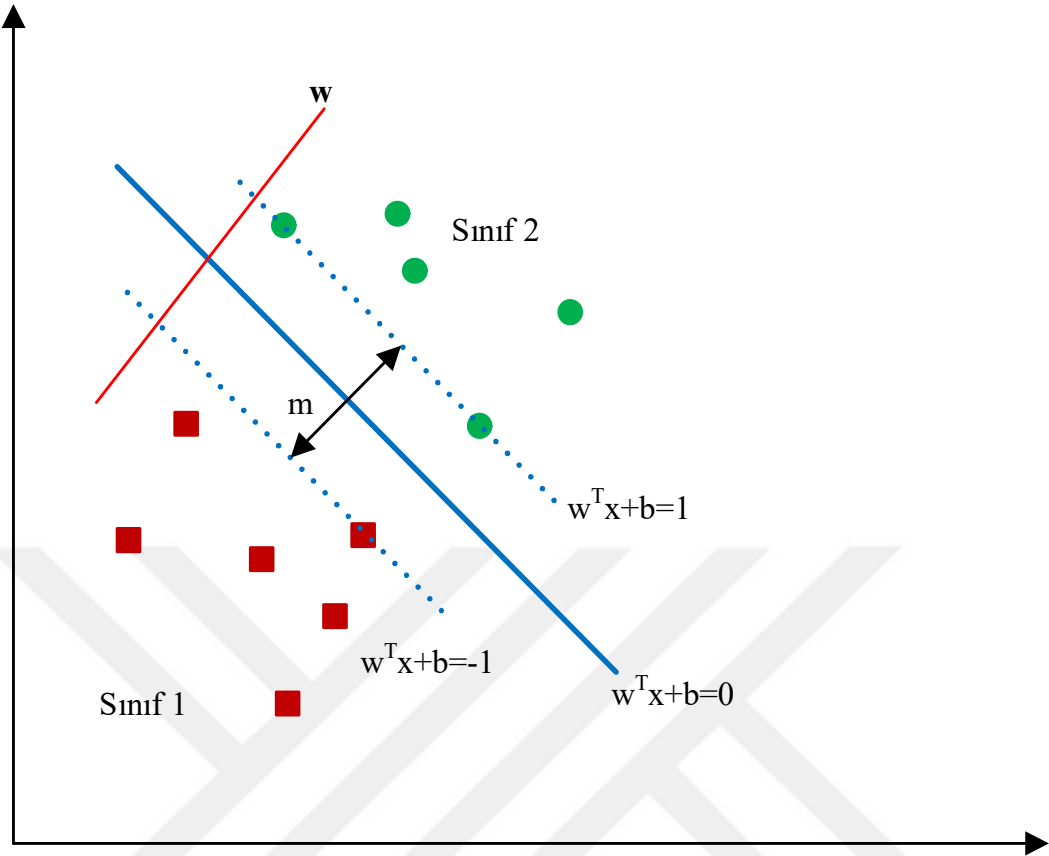
Burada w ; dizinin ağırlık vektörü, b ise skaler bir sabittir. Veri kümesini birbirinden ayıracak birçok hiper düzlem bulunmaktadır. Fakat iki düzlem arasında maksimum uzaklıktaki sınırları seçmek en uygun olanıdır [61]. Bu durumda denklem aşağıdaki gibi yazılabilmektedir:

$$\begin{aligned} w^T x + b &\geq 1 \quad \text{ise} \quad y_i = 1 \\ w^T x + b &\leq -1 \quad \text{ise} \quad y_i = -1 \end{aligned} \quad (3.36)$$

Bu durumda doğrusal sınıflandırma aşağıdaki gibi yazılabilmektedir:

$$f(x) = \text{sign}(w^T x + b) \quad (3.37)$$

İki sınıflı bir veri kümesi için hiper düzlemin yeri maksimum marjlinli (kenar aralığı) DVM yapısı Şekil 3.15'te gösterilmiştir.



Şekil 3.15. İki sınıflı veri kümesi için en uygun hiper düzlemin yerinin gösterimi

Belirli veri kümesi ve karar düzlemi için bir düzlemle i 'inci örnek ile x_i 'nin fonksiyonel marjini (w, b) aşağıdaki gibi tanımlanmaktadır:

$$\gamma_i = y_i(w^T x_i + b) \quad (3.38)$$

Bir düzlem ve bir nokta arasındaki bilinen en kısa mesafe, düzleme dik ve dolayısıyla w 'ya paraleldir. Bu yöndeki bir birim vektör $w/\|w\|$ 'dur. Bu durumda denklem (3.38)'deki ifade aşağıdaki gibi yazılmaktadır:

$$\gamma_i = \frac{y_i(w^T x_i + b)}{\|w\|} = \frac{1}{\|w\|} \quad (3.39)$$

Lineer ayırıcı tasarımın maksimize etmek için bir önceki denklemdeki geometrik marjinde en iyi w ve b 'yi bulmak için aşağıdaki denklemden faydalanılmaktadır:

$$y_i(w^T x_i + b) \geq 1 \quad (3.40)$$

Bu denklemin minimizasyonu aşağıdaki ifade ile gösterilmiştir:

$$\mathcal{O}(w) = w = w^T w \quad (3.41)$$

İki sınıfın geniş bir marjın boyunca net olarak ayıramadığı durumlarda sınır aşmalarındaki hatayı simgeleyen ζ_i parametresi bir önceki denkleme ilave edilmektedir:

$$\min \mathcal{O}(w) = \frac{1}{2} w^2 + C \sum_{i=1}^N \zeta_i \quad (3.42)$$

burada C ; düzenleme faktörünü ifade etmektedir.

3.5. Evrimsel Sinir Ağları

Son zamanlarda, en sık kullanılan derin öğrenme yöntemlerinden biri olan ESA mimarileri, nesne tanıma, yüz bilgisi çıkarma gibi bilgisayarlı görme alanındaki üstün performansı nedeniyle oldukça popülerlik kazanmıştır [62-64]. ESA, toplanmış bilgileri uçtan uca öğrenebilen bir mimaridir [65]. Tanımlama özelliklerini ham verilerden türetir ve bu da özellik çıkarmadaki insan gücünün azaltılması demektir. Aynı zamanda ESA, yeterli kapasitesi ve akıllı model yapısı sayesinde büyük ölçekli verilerin de üstesinden gelebilmektedir. Dolayısıyla ESA, hem özellik çıkarma hem de sınıflandırmayı bir arada barındıran önemli bir mimari olarak literatürdeki yerini almıştır [66,67]. ESA'nın bu avantajından dolayı üstün performans elde edilmektedir.

Bir ESA, bireysel nöronların, görsel alanın üst üste gelen bölgelerine karşılık gelecek şekilde döşenen ileri beslemeli bir yapay sinir ağı türüdür. Hubel ve Wiesel'in 1968'de kedilerdeki görsel korteks (beyindeki ince doku katmanı) üzerindeki ilk çalışmalarından, görsel korteksin karmaşık bir nöron dizilimi içerdiği bilinmektedir [68]. Bu nöronlar görsel alanın küçük alt bölgelerine duyarlıdır. Alt bölgeler, tüm görsel alanı kaplayacak şekilde döşenmiştir. Nöronlar bilgi alanı üzerindeki yerel filtreler olarak çalışırlar ve görüntüde bulunan yerel korelasyondan yararlanmak için oldukça uygundur.

Fukushima'nın çalışması nöronlar arasındaki bu yerel bağlanmalara ve görüntünün hiyerarşik olarak düzenlenmiş dönüşümlerine dayanan modelleri hesaplamaktadır [69]. Bu çalışmada aynı parametrelere sahip nöronlar, farklı lokasyonlarda bir önceki tabakanın yamalarına uygulandığında, bir dizi dönüşümsel değişmezlik elde edildiğini keşfetmiştir. LeCun ve arkadaşları da bu fikirden yola çıkarak doku tanımadaki performansı artırmak için geri yayılım algoritması ile hata eğrisini kullanarak ESA'yı tasarlamış ve eğitmiştir [70,71]. ESA'nın sinir ağı alanındaki ana katkılarından biri, aynı özellik haritasında aynı ağırlıkları uygulayarak ağırlık paylaşımının uygulanmasıdır. Bu da eğitilebilir parametrelerin sayısını belirgin bir şekilde azaltarak öğrenme verimliliğini artırmıştır.

Son yıllarda ESA mimarisine özellik haritalarının doğrusalsızlığını ve bağımsızlığını artıran aktivasyon fonksiyonları ve bırakma algoritması da uygulanmıştır [72]. Bu sayede daha akıllı ve kararlı öğrenilebilen özellikler kazandırılmıştır. Uçtan uca bir yapı olarak uygulanmasının yanında bazı araştırmalar, yapı içindeki derin öğrenme özelliklerinin, bilgisayarlı görme görevlerini yerine getirmek için sınıflandırıcılara giren özellikler olarak da kullanılabileceğini göstermektedir [73]. Kısacası ESA yapısı hem özellik çıkarma hem de sınıflandırıcı olarak kullanılmaktadır.

Geleneksel doku tanıma ve bilgisayarlı görme algoritmaları ile karşılaştırıldığında ESA mimarisi daha az ön işlem gerektirmektedir. Başka bir deyişle ağ, geleneksel algoritmalarda manuel olarak tasarlanmış filtreleri öğrenmekten sorumludur. Mevcut filtrelerin özelliklerini tasarlama zorluğu ile karşılaştırıldığında ön bilgi bağımsızlığı ESA'nın önemli bir avantajıdır.

3.5.1. ESA Yapısı

Bir ESA'da iki temel katman türü mevcuttur. Bunlardan biri evrişim diğeri de havuzlama katmanıdır. ESA'nın geliştirilmesi ile birlikte performansı artırmak için aktivasyon fonksiyonları ve bırakma işlemleri de eklenmiştir. ESA'nın son katmanında ise görev türüne göre farklı kayıp katmanlar seçilmiştir. Bu tez çalışmasının bundan sonraki alt başlıklarında ESA mimarisinde bulunan katmanlardan detaylı bir şekilde bahsedilmiştir. Şekil 3.16'da ESA'nın genel yapısının bir örneğini göstermektedir.

konumlarından özellik tipleri ile etkinleştirilecek filtreleri öğrenmektedir. Ardından giriş görüntüsü tüm filtreler için derinlik boyutu kadar istiflenerek aktivasyon haritalarının çıktısını oluşturmaktadır. Ağırlık paylaşımının sağladığı fayda ile girdi verilerinden daha fazla bilginin çıkarılmasını sağlanmış olmaktadır.

Evrişim katmanındaki bir özellik haritası, görüntünün alt bölgeleri boyunca tekrar tekrar uygulanmasıyla elde edilmektedir. Filtresi W^k ve biası b^k olarak ayarlanmış giriş haritasının h^k evriştirilecek katmanı olsun. Bu durumda h^{k+1} katmanındaki özellik haritasının çıkışı aşağıdaki denklem ile ifade edilmektedir:

$$h^{k+1} = W^k * h^k + b^k \quad (3.43)$$

Burada $*$ ifadesi evrişim işlemidir. Bu denklem tam bağlı katman için evrişim katmanının özel bir halidir. Tam bağlı katman, önceki katmandaki tüm nöronları alıp, sahip olduğu her nörona bağlayan bir evrişim katmanıdır. Diğer bir deyişle ağırlık paylaşımı olmayan bir evrişim katmanıdır. Ayrıca bu katmanın her bir filtresi 1×1 boyutundadır. Tam bağlı katmanın temel işlevi nöronların uzamsal olarak konumlandırılmasını azaltmak ve ileri derecede anlama özelliklerini oluşturmaktır.

3.5.1.2. Havuzlama Katmanı

Bir ESA mimarisinde havuzlama katmanı, çoğunlukla evrişim katmanlarının arasına periyodik olarak yerleştirilmektedir. Böylece aşırı uyum problemini bastırmasıyla uzamsal değişmezlik elde edilmiş olmaktadır [76]. Bir havuzlama katmanında her havuzlanmış özellik haritası, önceki katmandaki özellik haritasına karşılık gelmektedir. Şekil 3.16'da gösterildiği gibi küçük bir $n \times n$ 'lik yama, özellik haritasının birimlerini birleştirmek için kullanılmaktadır. Bu sayede daha büyük yerel alanlar üzerinde konum değişmezliği oluşturulmaktadır. Aynı zamanda girişi her yönde $n \times n$ faktörü ile aşağı örneklemektedir [77]. Aşağıdaki denklem ESA'daki genel havuzlama işlemini göstermektedir. Filtre boyutu F ve adım sayısı S olsun ve veri girdi boyutu $X \times Y \times Z$ kabul edilsin. $\hat{X} \times \hat{Y} \times \hat{Z}$ boyutlu bir çıkış özellik haritası üretmek için:

$$\begin{aligned}
\hat{X} &= \frac{X - F + S}{S} \\
\hat{Y} &= \frac{Y - F + S}{S} \\
\hat{Z} &= Z
\end{aligned} \tag{3.44}$$

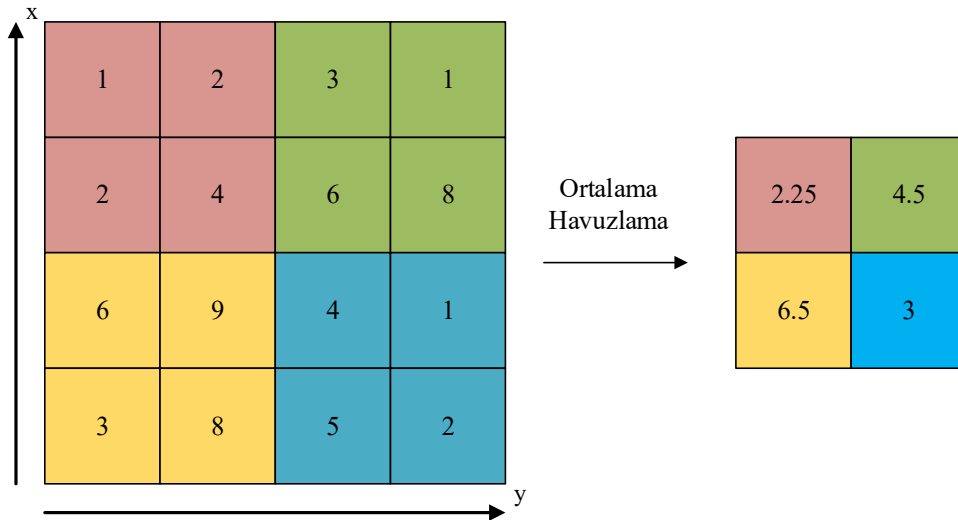
formülleri kullanılmaktadır. ESA’da kullanılmak üzere alt örnekleme ve maksimum olan iki temel havuzlama katmanı mevcuttur.

3.5.1.2.1. Alt Örnekleme Havuzlama

Bir alt örnekleme havuzlama için işlem aşağıdaki denklem ile gösterilmektedir:

$$a_j = \beta \sum_{N \times N} a_i^{n \times n} + b \tag{3.45}$$

burada a_j ; havuzlama katmanının çıkışı, a_i ise havuzlama katmanının girişidir. Alt örnekleme havuzlama işlemi, girdilerin toplamını alır, eğitilebilir bir β değeri ile çarpıp b biası ile toplanarak elde edilebilmektedir. ESA’da yaygın olarak kullanılan havuzlama, alt örneklem havuzlamanın özel bir durumudur ve $\beta = 1/N \times N$ ve $b=0$ kabul edilmektedir. Alt örnekleme havuzlama katmanı için bir örnek Şekil 3.18’de gösterilmiştir.



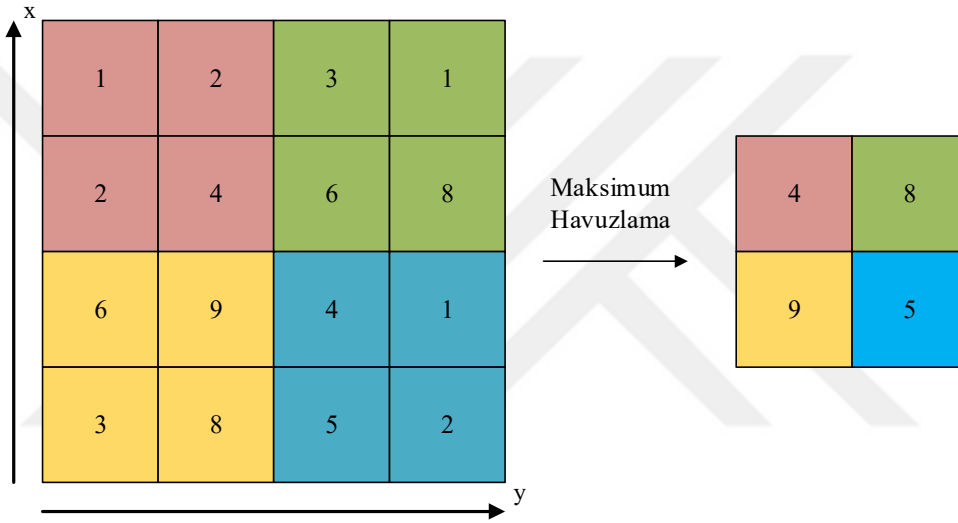
Şekil 3.18. Ortalama havuzlama örneği

3.5.1.2.2. Maksimum Havuzlama

Bir maksimum havuzlama için işlem aşağıdaki denklem ile gösterilmektedir:

$$a_j = \max(a_i^{n \times n} u(n, n)) \quad (3.46)$$

Girilen verilere bir $u(x, y)$ pencere fonksiyonu uygulanarak komşuluktaki maksimum değerleri çıkarmaktadır. Şekil 3.19'da maksimum havuzlama işleminin bir örneği gösterilmektedir.



Şekil 3.19. Maksimum havuzlama örneği

Ortalama havuzlama geleneksel ESA'da yaygın olarak kullanılmasına rağmen maksimum havuzlama deneylerde daha iyi performans göstermektedir ve ESA mimarisinde oldukça yaygın bir şekilde kullanılmaktadır. Scherer ve arkadaşları nesne tanıma ile ilgili çalışmalarında alt örnekleme ve maksimum havuzlama performansını karşılaştırmak için bir deney gerçekleştirmiştir [76]. Deney sonucunda maksimum havuzlamanın daha üstün olduğu ispatlanmıştır. Ayrıca maksimum havuzlama, genelleştirmedeki performansı artıran ve eğitilebilir parametrelerin sayısını azaltan üstün değişmez özellikler seçerek daha hızlı yakınsama oranına olanak sağlamaktadır. Böylece hesaplamaları minimuma çekerek bilgi işleme zamanını da azaltmış olmaktadır [78].

3.5.1.3. Aktivasyon Fonksiyonu

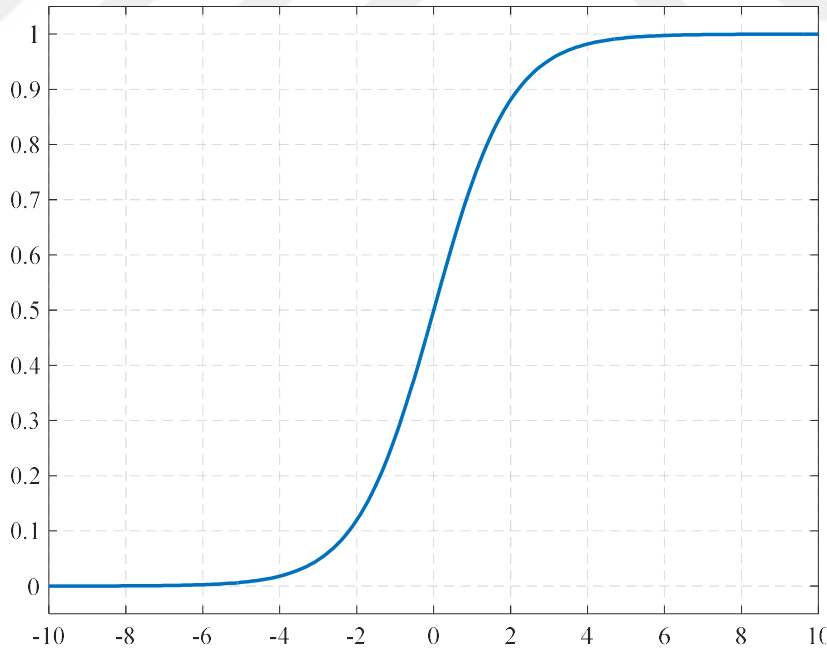
ESA yapılarında en büyük faktörlerden biri de aktivasyon fonksiyonunun uygulanmasıdır. Aktivasyon fonksiyonları, giriş yığınının yüksek düzeyde anlaşılmasını sağlayan ağırlardaki doğrusalsızlıkları artırmaktadır. Ayrıca aktivasyon fonksiyonu, aşırı veri değerleri olmadan bir sonraki katmandaki nöronların bağımsızlığını artıran bir özellik haritası çıkarmaktadır. Sonuç olarak bu sayede tüm ağıın istikrarı artırılmaktadır.

Bu alt bölümde ESA mimarisinde kullanılan aktivasyon fonksiyonlarından bahsedilecektir.

3.5.1.3.1. Lojistik Sigmoid Fonksiyonu

Sigmoid fonksiyonu aşağıdaki eşitlik ile tanımlanmaktadır:

$$\text{sigm}(x) = \frac{1}{1 + e^{-x}} \quad (3.47)$$



Şekil 3.20. $\text{sigm}(x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için)

Şekilden görüleceği üzere sigmoid fonksiyonu reel değerler almaktadır. Diğer bir deyişle ilgili piksel değerlerini 0 ile 1 aralığına bastırmaktadır. Özellikle büyük negatif değerler 0, büyük pozitif değerler ise 1 olma eğilimindedirler.

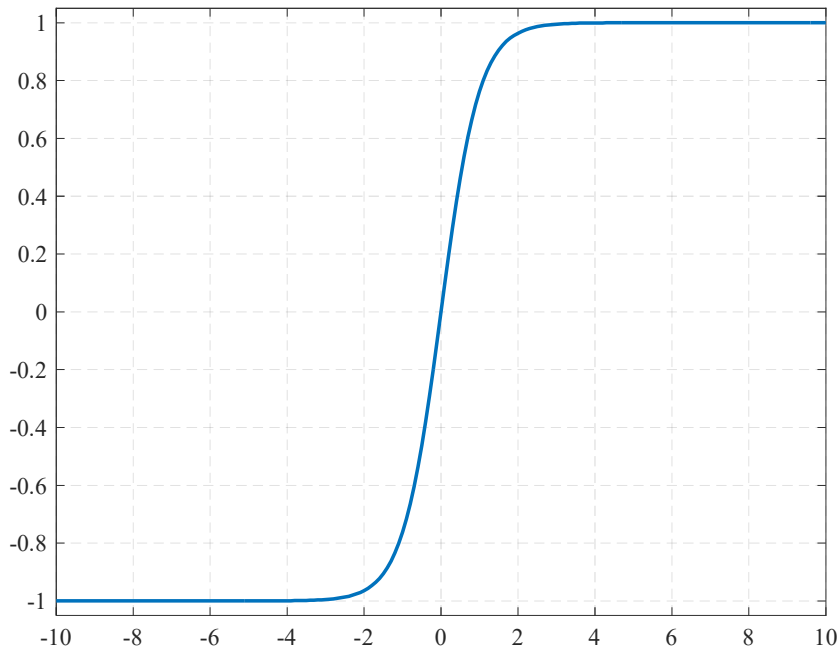
Sigmoid fonksiyonu yaygın olarak kullanılmasına rağmen son araştırmalar [79], önemli bir dezavantajından dolayı bazı uygulamalarda iyi performans göstermediğini ispatlanmıştır. Bu fonksiyon kolaylıkla doyuma ulaştırılmaktadır ve eğitim sürecinde gradyanları bastırmaktadır. Fonksiyonun aktivasyonu 0'a ya da 1'e doymuş olduğunda bu bölgelerdeki gradyanlar neredeyse 0 olmaktadır [79]. Bu da arzu edilmeyen bir durum teşkil etmektedir.

3.5.1.3.2. Hiperbolik Tanjant Fonksiyonu

Hiperbolik tanjant fonksiyonu aşağıdaki eşitlik ile tanımlanmaktadır:

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.48)$$

Bu fonksiyonun grafiği şekilde gösterilmiştir. Sigmoid fonksiyonuna benzer şekilde tanjant fonksiyonu doğrusal olmayan bir şekilde $[-1, 1]$ sayı aralığındaki bir reel değere bastırmaktadır. Bu fonksiyonunda da doymuş aktivasyon problemi mevcuttur. Tanjant fonksiyonu sigmoid fonksiyonundan daha başarılı olmasına rağmen, uygulamada hala iyi performans göstermemektedir.



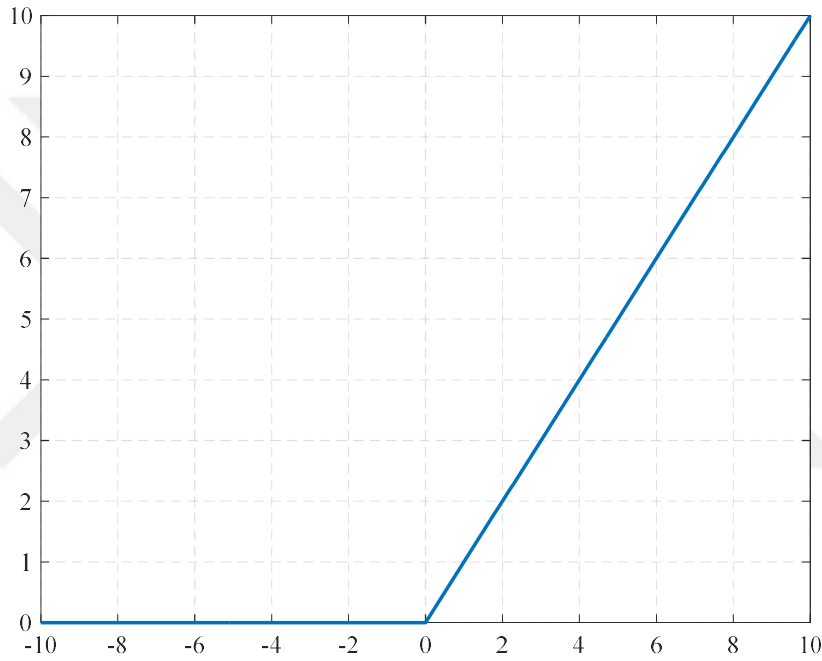
Şekil 3.21. $\tanh(x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için)

3.5.1.3.3. Düzeltilmiş Doğru Birimi

Bu aktivasyon son zamanlarda ESA mimarisinde en çok kullanılan fonksiyondur ve aşağıdaki denklem ile ifade edilmektedir:

$$f(x) = \max(0, x) \quad (3.49)$$

Bu fonksiyonun grafiği Şekil 3.22’de gösterilmiştir.



Şekil 3.22. $\max(0, x)$ fonksiyonunun grafiği ($-10 < x < 10$ aralığı için)

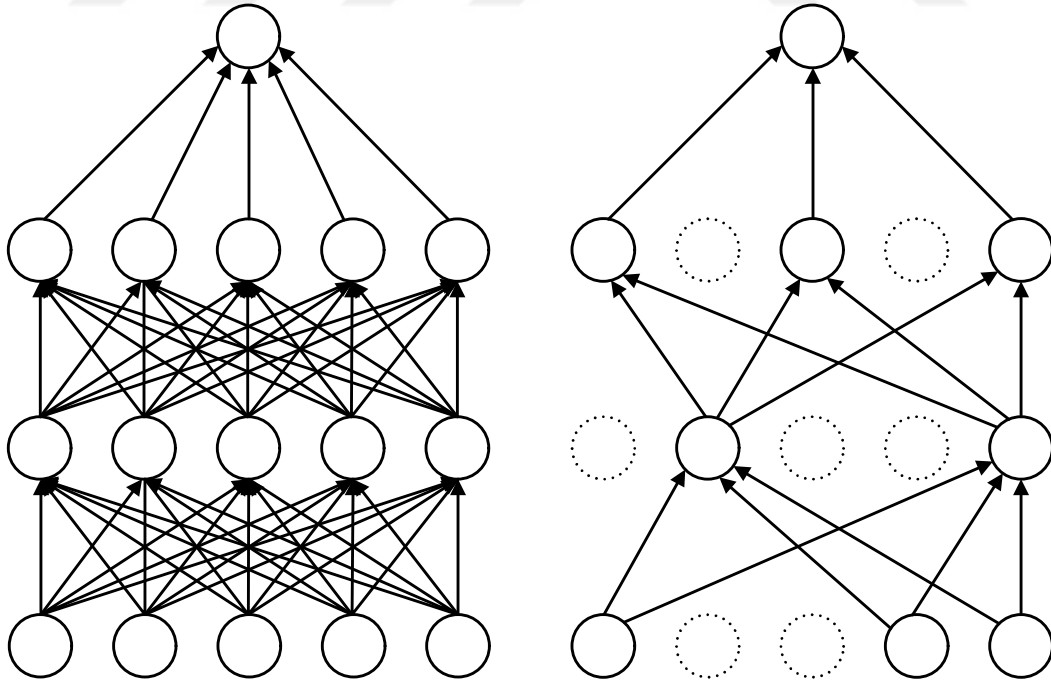
Düzeltilmiş doğrusal birim fonksiyonu iki temel avantaja sahiptir. İlk olarak, karmaşık işlemler yani üstel ifadeler içeren sigmoid fonksiyonu ve $\tanh$ fonksiyonu ile karşılaştırıldığında, basitçe bir eşiği sıfıra ayarlamak suretiyle uygulanmaktadır. Böylece düzeltilmiş doğrusal birimli ESA mimarisi, $\tanh$ ve sigmoid fonksiyonlu eşdeğerlerinden birkaç kat daha hızlı eğitilmektedir [62]. İkinci olarak ESA’nın çok fazla ön işleme ihtiyacı olmamasıdır.

Bununla birlikte, bu fonksiyon için de bir dezavantaj vardır. Öğrenme oranı çok büyük bir değere ayarlanırsa verinin yapısı bozulacağından, performansa doğrudan etki etmektedir. Fakat bu durum daha az sıklıkta öğrenme oranının uygun bir şekilde ayarlanmamasıyla ilgili bir sorundur.

3.5.1.4. Bırakma

Bir ESA mimarisi, girdiler ve çıktılar arasındaki karmaşık ilişkileri öğrenebilen etkileyici bir model olan çoklu doğrusal olmayan gizli katmanlardan oluşmaktadır. Bununla birlikte sınırlı eğitim verisinin bulunduğu durumlarda test verileri aynı ağ ile işlenmiş olsa dahi eğitim veri setindeki gürültüden kaynaklanan bir karmaşıklık oluşacaktır [80]. Ardından da eğitim süreci sırasında aşırı uyum meydana gelecektir. Bu sorunu azaltmak için doğrulama aşamasında performans kötüleşmeye başlar başlamaz eğitimin durdurulması, çeşitli ağırlık türlerini eleme ve hafif ağırlık paylaşımı gibi farklı yöntemler geliştirilmiştir [81].

Bırakma (dropout), büyük sinir ağının genel hatasını azaltan aşırı uyum problemini çözmek için sunulan güçlü bir algoritmadır [72]. Bir bırakma algoritmasında tek bir nöron diğer nöronların oluşumlarına güvenmeyeceği için nöronların uyum karmaşıklıklarını azaltır. Böylece bırakma, daha sağlam özellikler ve istikrarlı yapıyı öğrenebilmek için ESA'yı geliştirmiştir [82]. Bu durumda bırakma terimi bir sinir ağındaki birimlerin bazılarını bir sonraki katmana dahil etmeme anlamına gelmektedir. Başka bir deyişle bir birimi dışarı atmak, birimi tüm bağlantılarıyla geçici olarak ağdan kaldırmak olarak ta ifade edilebilmektedir [72].



Şekil 3.23. Basit bir bırakma örneği. Sol: iki gizli katmana sahip standart bir sinir ağı. Sağ: Sol taraftaki ağa bırakma uygulanarak üretilen inceltilmiş bir ağ örneği

Şekil 3.23'ten de görüleceği üzere standart sinir ağı ile kıyaslandığında bırakma yöntemine sahip ağ, eğitilebilir parametrelerin sayısını büyük ölçüde azaltmaktadır. Böylece

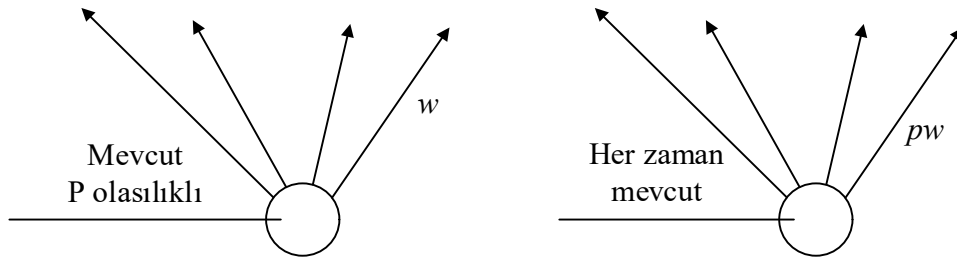
hesaplamaların sayısı azaltılmaktadır ve hesaplama verimliliğini artırılmaktadır. Pratikte ESA, Şekil 3.23'teki örnekten daha büyük ve derindir. Ayrıca bırakma yönteminin performansa etkisi oldukça belirgindir.

Bırakma, bir ESA'yı düzene koymak için gizli birimlere gürültü ekleyen bir yöntem olarak ta yorumlanabilmektedir. Birimlerin durumlarına gürültü ekleme fikri daha önceden gürültü gidermede kullanılmaktadır [83]. Otomatik enkoderin giriş ünitelerine gürültü eklenir ve ağ, gürültülü girişin yeniden yapılandırılması için eğitilir. Ancak gürültü eklemeden farklı olarak bırakma, kayıp katman dışındaki tüm katmanlarda kullanılır ve uçtan uca geri yayımlı denetlenen eğitim boyunca gerçekleştirilir. Hangi birimlerin bırakılacağı rastgele seçilir. Bir evrişim katmanlı bırakmanın ileri işlemi aşağıdaki denklemde gösterilmiştir:

$$h^{k+1} = M \otimes (W^k * h^k b^k) \quad (3.50)$$

Burada M ; bırakma maskesini ve $\otimes$ ise; eleman elemana çarpımı ifade etmektedir. Bırakma oranı p olarak ayarlanır ($0 < p < 1$) ve M 'nin her bir elemanı, p olasılığını sıfıra eşitler (ayrıca olasılık $1-p=1$ olur). Bu nedenle M ikili bir maskedir.

Test etme sırasında, eğitim sürecinde ortaya çıkan tüm bırakma durumları ile tahminlerden ortalama alınması uygun bir durum değildir. Bu problemi çözmek için basit bir yaklaşık ortalama yöntemi uygulanmaktadır ve iyi çalışmaktadır. Bu fikirde amaç eğitimden geçmiş ağı test sırasında bırakmadan kullanmaktır. Bu ağı ağırlıkları, eğitilmiş ağırlıkların küçültülmüş bir versiyonudur. Bir ünite eğitim sürecinde olasılık p ile eğitilirse, o birimin çıkış ağırlıkları test sırasında p ile çarpılır. Bu durum Şekil 3.24'te gösterilmiştir.



Şekil 3.24. Sol: eğitim sürecinde bir birim. Sağ: Test sürecinde bir birim. Eğitim sürecindeki birim, olasılık p ile sunulur ve eğitilmiş w ağırlığı ile bir sonraki katmanda birimlere bağlanır. Test sürecinde, Birim her zaman mevcuttur ve ağırlıklar p ile çarpılır. Test sürecinin çıktısı, eğitim sürecinin beklenen çıktısıyla aynıdır

Bu ölçekleme ile herhangi bir gizli ünite için beklenen çıktı, test sırasındaki gerçek çıktı ile aynı olmaktadır. Kaynak [84]'te bir ağı bırakma ile eğitilmesinin ve test zamanında bu

yaklaşık ortalama yöntemin kullanılmasının diğer düzenleyici yöntemlerle karşılaştırıldığında çeşitli sınıflandırma problemlerinde genelleme hatasını belirgin bir şekilde düşürdüğü gösterilmiştir.

3.5.1.5. Kayıp Fonksiyonu

ESA’lardaki farklı görevler için farklı kayıp fonksiyonlar seçilmektedir. Bu alt bölümde yaygın olarak kullanılmakta olan iki kayıp fonksiyondan bahsedilecektir.

3.5.1.5.1. Öklid Fonksiyonu

Öklid kaybı, gerçek değerli regresyon görevleri için kullanılmaktadır. Tek bir reel sayı olduğu için Öklid kaybı ile ESA’nın son katmanını 1×1 boyutlu olmaktadır. Öklid kaybının matematiksel fonksiyonu aşağıda gösterilmektedir.

$$L = \frac{1}{2N} \sum_{i=1}^N \|\hat{d}_i - d_i\|_2^2 \quad (3.51)$$

Burada $\hat{d}_i$; indirgenmiş çıkışları, d_i ; hedef çıkışları ve N ise; çıkış sayısını göstermektedir.

3.5.1.5.2. Softmax Fonksiyonu

Softmax kaybı, karşılıklı olarak K kadar sınıfı tahmin etmek için kullanılır ve $1 \times k$ boyutlu olasılık vektörünü çıkarır. Vektöre ait tüm olasılıkların toplamı bire eşit olmaktadır. Softmax kayıplarının matematiksel fonksiyonu aşağıda gösterilmektedir.

$$L = -\sum_j y_j \log(p_j) \quad (3.52)$$

Burada y_j ; j ’inci sınıfa ait hedef $y_j = 1$ olduğunda kesin doğrudur. Aksi halde $y_j = 0$ ’dır. p_j , girdinin j ’inci sınıfına ait tahmin edilen olasılığını belirtir. Tahmin edilen olasılık

vektörü çıkarıldığında, softmax fonksiyonunun matematiksel fonksiyonu aşağıdaki gibi olmaktadır.

$$p_j = \frac{e^{o_j}}{\sum_k e^{o_k}} \quad (3.53)$$

Burada o_j ; ESA'nın son katmanındaki j 'inci çıktıyı belirtmektedir. Softmax kaybı sınıflandırma görevleri için tasarlanmış olmakla birlikte regresyon görevleri üzerinde de uygulanabilmektedir. Bu tez çalışmasında son katman olarak softmax fonksiyonu kullanılmıştır ve çıkış katmanındaki olasılık vektör boyutu 1×2 'dir.

Sonuç olarak ESA mimarisi kendi kendine öğrenme kabiliyetine sahip bir derin öğrenme yapıdır. Bu kabiliyet, ağıın içerisindeki katman sayısını artırarak elde edilebilecek doğrusalsızlıklar ile artmaktadır [85].

3.6. Performans Ölçütleri

Bu alt bölümde sınıflandırıcıların performansını değerlendirmek amacıyla sıklıkla kullanılan ölçütlerden bahsedilmiştir.

3.6.1. Karışıklık Matrisi

Makine öğrenmesi ve istatistiksel öğrenme alanlarında karışıklık matrisi, bir başka ifadeyle hata matrisi ya da ikili ihtimal tablosu özel bir tablodur. Bu tablo, genellikle bir sınıflandırma başarısını ölçmek üzere kullanılmaktadır. Karışıklık matrisi, her bir satırında algoritmanın doğru ve yanlış şekilde tahmin ettiği örneklerle birlikte gerçek değerlerin eşleşmesi doğrultusunda elde edilen dört farklı sayısal endekslerden oluşmaktadır. Doğru pozitif (DP) ve doğru negatif (DN) sırasıyla doğru şekilde tahmin edilmiş doğru ve yanlış örneklerin sayısını temsil ederken; yanlış pozitif (YP) ve yanlış negatif (YN) sırasıyla yanlış şekilde tanımlanmış doğru ve yanlış örneklerin sayısını temsil etmektedir. Karışıklık matrisinin kullanımı, özellikle sınıflandırmanın olduğu uygulamalarda oldukça yaygın bir şekilde gözlemlenmektedir. Şekil 3.25'te karışıklık matrisine ait bir örnek gösterilmiştir. DP, DN, YP ve YN indislerinin temsil ettiği sayılar sırasıyla A , B , C ve D değerleri ile gösterilmiştir. Benzer şekilde, gerçek pozitif (GP) ve gerçek negatif (GN) ile pozitif tahmin

(PT) ve negatif tahmin (NT) değeri satır ve sütunların toplamlarından oluşacak şekilde $A+C$, $B+D$ ve $A+B$, $C+D$ değerleriyle gösterilmiştir [86].

	Hedef Sınıfı		
Çıkış Sınıfı	GP	GN	
	DP	YP	PT
	YN	DN	NT

	Hedef Sınıfı		
Çıkış Sınıfı	A+C	B+D	
	A	B	A+B
	C	D	C+D

Şekil 3.25. Karışıklık matrisi

Karışıklık matrisi sağladığı indisler sayesinde sınıflandırıcı karakteristiğini temsil eden çeşitli başarımların elde edilmesini sağlamaktadır. Bu kapsamda en sık kullanılan başarımların matematiksel formları aşağıda tanımlanmıştır.

Doğruluk ölçütü, sistemin tüm örnekler üzerindeki başarımlarını göstermektedir ve aşağıdaki denklem ile tanımlanmaktadır:

$$\text{Doğruluk} = \frac{DP + DN}{DP + YP + YN + DN} = \frac{A + D}{A + B + C + D} \quad (3.54)$$

Duyarlılık ölçütü, sistemin pozitif örnekler üzerindeki başarımlarını ölçmek üzere kullanılmaktadır ve aşağıdaki denklem ile gösterilmektedir:

$$\text{Duyarlılık} = \frac{DP}{DP + YN} = \frac{A}{A + C} \quad (3.55)$$

Özgüllük ölçütü, sistemin negatif örnekler üzerindeki başarımlarını ölçmek üzere kullanılmaktadır ve aşağıdaki denklem ile gösterilmektedir:

$$\text{Özgüllük} = \frac{DN}{DN + YP} = \frac{D}{D + B} \quad (3.56)$$

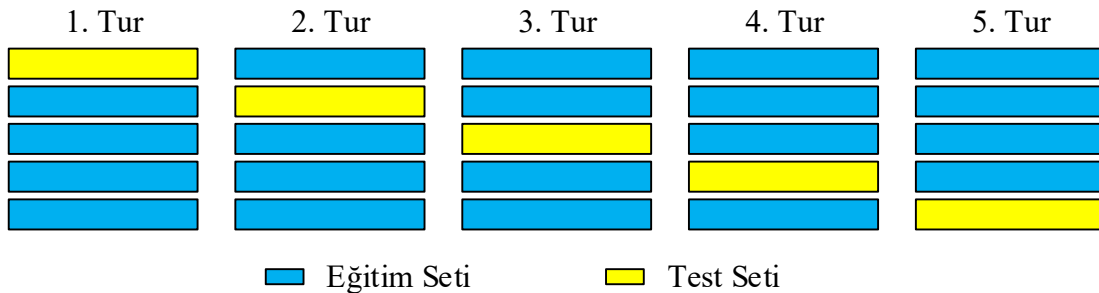
Doğruluk ile birlikte hata oranı geleneksel olarak kullanılan bir ölçümdür ve aşağıdaki denklem ile ifade edilmektedir:

$$Hata_Oranı = 1 - Doğruluk \quad (3.57)$$

Yukarıdaki performans ölçütlerinin verilerin sınıflar üzerinde dengeli dağıldığı durumlarda kullanılması uygun olacaktır. Bazı durumlarda, veri setinde sınıflandırılması gereken kayıtlar sınıflar üzerine eşit bir şekilde dağılmayabilir. Örneğin 150 doğru ve 50 yanlış havaalanı örneğinden oluşan bir veri setinde kayıtlar sınıflar üzerinde dengesiz bir şekilde dağılmıştır. Verilerin sınıflar üzerinde dengesiz bir şekilde dağıldığı durumlarda sınıflandırıcılar, çoğunluğu oluşturan sınıfı ayırt etme konusunda yanlış davranabilir. Bu durumda, veri setinde azınlığı oluşturan sınıfın ayırt edilmesi, çoğunluğa göre daha zayıf kalabilir [87]. Dolayısıyla çoğunluğu oluşturan sınıfa bağlı olarak, çok yüksek duyarlılık değeriyle birlikte düşük özgüllük ya da çok düşük özgüllük değerlerinin elde edilmesi olasıdır.

3.6.1. *k*-Katlı Çapraz Doğrulama

Çapraz doğrulama, iyi bir model değerlendirme tekniğidir. Modeller değerlendirilirken modelin daha önce görmediği veriler üzerinde bir tahmin yapılması istendiğinde, modelin bunu ne kadar iyi yapacağı konusunda herhangi bir bilgi yoktur ve bu bir sorundur [88]. Bu sorunu aşmanın yollarından biri, modeli eğitirken verilerin tamamının eğitim için kullanılmamasıdır. Bu nedenle, verilerin bir kısmı eğitimden önce veri setinden çıkartılır. Eğitim tamamlandıktan sonra veri setinden çıkartılan örnekler modeli test etmek üzere kullanılır. Çapraz doğrulama denilen model değerlendirme yöntemi tam olarak bu durumu ifade eder.



Şekil 3.26. 5-katlı çapraz doğrulama örneği

Hangi verilerin test için dışarıda bırakacağı veya bu verilerin nasıl seçileceği konusunda *k*-katlı çapraz doğrulama tekniği kabul görmüş bir yaklaşımdır. *k*-katlı çapraz doğrulamada, veri seti *k* alt parçaya bölünür ve dışarda tutma işlemi *k* defa gerçekleştirilmektedir. Her

seferinde, k alt kümelerinden biri test kümesi olarak kullanılırken diğer $k-1$ alt kümeleri bir eğitim kümesi oluşturmak üzere bir araya getirilmektedir. Daha sonra tüm k denemelerindeki ortalama hata hesaba katılmaktadır. Bu yöntem, verilerin nasıl bölündüğü konusunun daha az önemsenmesini sağlamaktadır. Her veri örneği tam olarak bir kez test kümesine girer ve $k-1$ defa eğitim kümesinde yer alır. Sonuç olarak tahmin varyansı, k değeri arttıkça azaltılır. Bu yöntemin dezavantajı ise eğitim algoritmasının sıfırdan k defa tekrarlanması gerekliliğidir, yani değerlendirme yapmak için k defa hesaplama gerektirir. Bu yöntemin bir başka türü, verileri rastgele bir test ve eğitim setine k farklı zamanlara bölmektir. Bunu yapmanın avantajı, her test kümesinin ne kadar büyük olduğunu ve kaç denemenin bağımsız olduğunu seçilebilmesidir [89].



4. DENEYSEL SONUÇLAR

Bu tez çalışmasında uydu görüntülerinden havaalanı tespit problemlerine iki farklı çözüm yöntemi önerilmiştir.

Önerilen birinci yöntemi test etmek için Google Earth'ten [90] elde edilen ve içerisinde havaalanları bulunan UAG'lerde bazı deneyler yapılmıştır. Tüm görüntüler gri ölçeklere dönüştürülmüş ve boyutları 3000×2000 olan 1 metre çözünürlüğe sahip, içerisinde hava alanı olan görüntüleri kullanılmıştır. Deneyler MATLAB 2015b'de Intel Core i7-3630QM 3.2GHz işlemciye ve 8GB hafızaya sahip bir bilgisayarda gerçekleştirilmiştir.

Toplamda 92 veri setinden oluşan görüntülerden 48'i eğitim ve geriye kalan 44 görüntü ise test için kullanılmıştır. Eğitim ve test veri kümesi rastgele seçilmiştir. Eğitim görüntüleri için geliştirdiğimiz DPA yöntemi aracılığıyla 67 havaalanı bölgesi ve 271 havaalanı olmayan bölge belirlenmiştir. Ayrıca test görüntüleri için de 50 havaalanı bölgesi ve 173 havaalanı olmayan bölge tespit edilmiştir. Kaynak [10]'da olduğu gibi eğitim ve test görüntülerindeki bu aday bölgeler, havaalanı (pozitif) ve havaalanı değil (negatif) olarak manuel olarak etiketlenmiştir. Ancak Kaynak [10]'dan farklı olarak %30'luk eşik değeri yerine %40'lık bir eşik değeri kullanılmıştır. Diğer bir deyişle tespit edilen ve bir dikdörtgen içerisine alınan aday havaalanların %40'ından fazlası bulunan bölge içerisinde ise pozitif, aksi halde negatif olarak etkilenmiştir. Veri setindeki tüm havaalanlarının geliştirdiğimiz DPA yöntemi ile aday bölgeler olarak bulunduğu unutulmamalıdır. Ancak bazıları önerdiğimiz yöntem ile doğrulanamamıştır. Önerilen yöntemin test veri seti üzerindeki performansını değerlendirmek için doğruluk, duyarlılık ve özgüllük kriterleri kullanılmıştır. Bölüm 3.1'de bahsedilen sekiz parametrenin havaalanı aday bölgelerinin tespitine etkisi kapsamlı deneylerle analiz edilmiş ve geliştirdiğimiz DPA yönteminin Tablo 4.1'de verilen aralıklarda oldukça gürbüz olduğu gözlenmiştir. Ardından parametre değerleri Tablo 4.1'de olduğu gibi ayarlanmıştır.

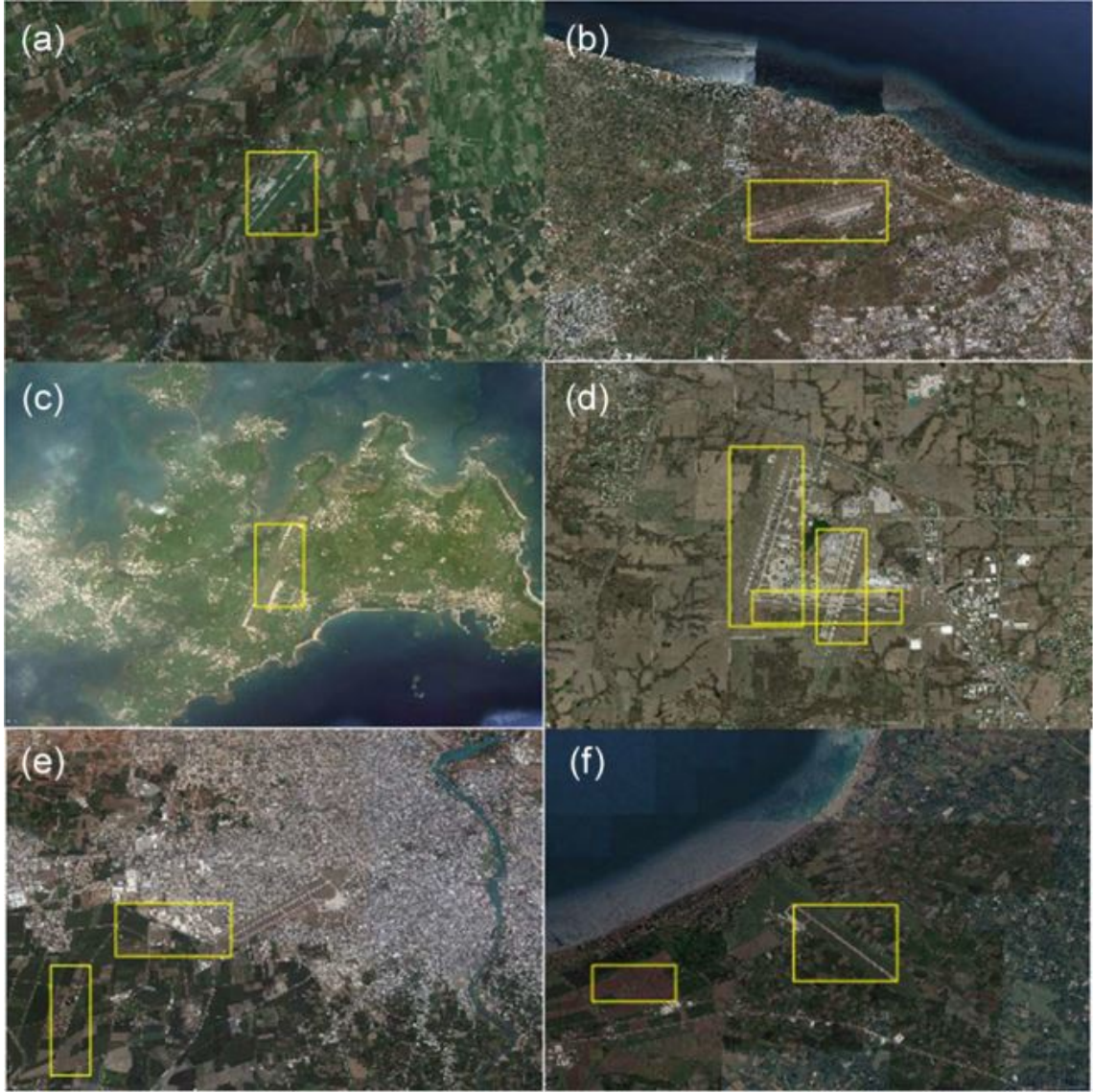
Tablo 4.1. Geliştirilen DPA için seçilen parametreler

Parametre		Denenmiş aralık	Seçilen Değer
thr_{θ}	(Adım 1)	[1-7]	1
thr_{d1}	(Adım 1)	[45-80]	55
thr_{d2}	(Adım 2)	[350-850]	450
thr_{d3}	(Adım 3)	[0,5-20]	3
thr_{d4}	(Adım 4)	[70-270]	150
thr_{d5}	(Adım 5)	[50-550]	100
Minimum uzunluk	(Adım 6)	[200-400]	300
Maksimum uzunluk	(Adım 6)	[900-1250]	1150

Ayrıca ÖBÖD tanımlayıcıları, aşağıdaki parametrelere göre ayarlanmıştır: Izgara noktası aralığı 10, destek bölgelerin dairesel yarı çapları 4, 6, 8 ve 10 olarak ayarlanmıştır. Eğitim veri seti, GKM kod kitabını inşa etmek için 25 karışımla kullanılmıştır. Ayrıca deneylerde farklı boyutlu karışımlar da denenmiştir. Ancak en iyi sonuç $N=25$ değeri için elde edilmiştir. Böylece $2 \times 128 \times 25 = 6400$ boyutlu FV üretilmiştir.

Hem homojen haritalama hem de FV kodlaması için VLFeat aracı kullanılmıştır [91]. DVM'nin etkinliğini artırmak için LIBLINEAR kütüphanesi de kullanılmıştır [92]. LIBLINEAR, geniş ölçekli doğrusal sınıflandırma için açık kaynak kütüphane olarak geliştirilmiştir.

Önerilen yöntemin karmaşık zeminler ve çeşitli havaalanı yapılarında gürbüz olduğunu kanıtlamak için başarıyla tespit edilen havaalanları Şekil 4.1(a)-(d)'de gösterilmektedir. Şekil 4.1(e) ve (f)'de ise yanlış-negatif ve yanlış-pozitif örnekler verilmiştir.



Şekil 4.1. Havaalanı tespit örneği

En iyi performans ölçütünü elde edebilmek için eğitim setinde beş kat çapraz doğrulama ile sınıflandırıcı parametreleri ayarlanmıştır. MATLAB’da bulunan DVM, tüm çekirdek için denenmiş ve en iyi sonuç doğrusal çekirdek ile elde edilmiştir. C parametresi 100 olarak seçilmiştir. LIBLINEAR DVM için düzenlenmiş-L2 ve kayıp çözücü-L2 seçilmiştir. Ayrıca LIBLINEAR için C parametresi 0.07 olarak ayarlanmıştır. Homojen haritalama için χ^2 çekirdeği kullanılmıştır. Test veri seti için önerilen yöntemin performansı Tablo 4.2’de gösterilmiştir.

Tablo 4.2. Önerilen birinci yöntemin performansı

Sınıflandırıcı yapısı	Doğruluk	Duyarlılık	Özgüllük
DVM	%82,5	%56,0	%90,2
LIBLINEAR DVM	%92,8	%74,0	%98,3
Homojen Haritalama ve LIBLINEAR DVM	%94,6	%78,0	%100,0

DeneySEL sonuçlar, homojen haritalama ve LIBLINEAR DVM kombinasyonunun en iyi doğruluğun elde edildiğini ortaya koymaktadır. Sınıflandırma doğruluğu %94,6'dır. Ayrıca duyarlılık ve özgüllük değerleri sırasıyla %78 ve %100'dür. En kötü sınıflandırma başarımları ise, %82,5'lik doğruluk, %56'lık duyarlılık ve %90,2'lik özgüllük değerleri ile doğrusal çekirdek yapısının kullanıldığı DVM olmuştur. Bu nedenle LIBLINEAR yapısının sınıflandırma performansını açık bir şekilde artırdığı anlaşılmaktadır. LIBLINEAR DVM yapısı için sonuçlar DVM'den yaklaşık %10 daha yüksektir. Ayrıca LIBLINEAR DVM yapısı üzerinde homojen haritalama ile başarımlar sonucunu yaklaşık %2 daha artırılmıştır.

Bu tez çalışmasında ayrıca önerilen yöntem, Kaynak [8-10]'da sunulan üç yöntem ile karşılaştırılmıştır. Kaynak [9]'daki yöntem çalışmalarında kenar algılama tabanlı havaalanı aday bölge çıkarma ve DVM sınıflandırıcısı kullandığı için kıyaslama ölçütü olarak seçilmiştir. Kaynak [8] ise havaalanı aday bölge çıkarmak için Hough dönüşüm ve dikkat çekerek haritasını kullandığı için seçilmiştir. Ayrıca yöntemde bizim önerdiğimiz ÖBÖD özneliklerini de kullanmıştır. Son olarak Kaynak [10]'daki yöntem de önerdiğimiz yönteme benzediği için karşılaştırmaya değer olduğu düşünülmüştür. Ancak önerdiğimiz yöntem, aday havaalanı bölgesi çıkarımı bakımından farklılık göstermektedir. Kaynak [10]'da yazarlar aday bölge tespiti için dikkat çekerek haritası ve DPA tabanlı yaklaşımını birleştirmişlerdir. İlgili doğruluk karşılaştırmaları Tablo 4.3'te gösterilmiştir.

Tablo 4.3. Önerilen birinci yöntem ile literatürdeki diğer yöntemlerin karşılaştırılması

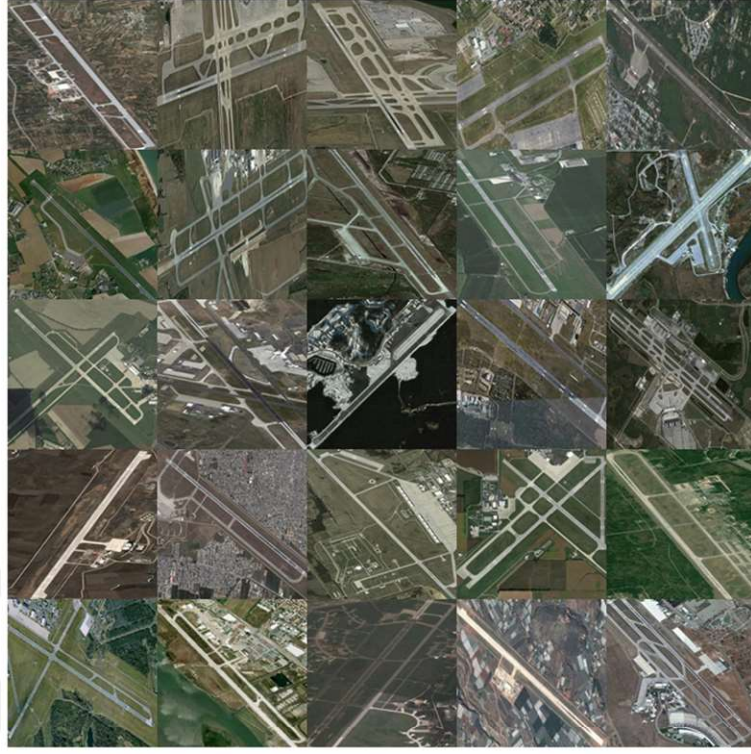
Yöntem	Doğruluk
<i>Önerilen Yöntem</i>	%94,6
<i>Wang ve diğ. [8]</i>	%83,8
<i>Qu ve diğ. [9]</i>	%70,8
<i>Qu ve diğ. [9] uygulanmış</i>	%71,7
<i>Zhu ve diğ. [10] uygulanmış</i>	%88,6

Önerilen yöntemin %94,6 doğruluk değeri ile diğer yöntemlerden daha iyi performansa sahip olduğu görülmektedir. Kaynak [10] ise ikinci en iyi doğruluğa sahiptir. Önerdiğimiz yöntemde kullanılan veri setleri ile Kaynak [8] ve [9]'da kullanılan veri setinin aynı olmadığı unutulmamalıdır. Diğer veri setleri mevcut olsaydı karşılaştırma daha uygun sayılabilirdi. Ancak burada yapılan karşılaştırmanın amacı önerdiğimiz yöntemin geçerliliği hakkında kabaca fikir vermektir. Bu sebeple kıyaslamamızın daha doğru yorumlanabilmesi için Kaynak [9] ve [10]'daki yöntemler kendi oluşturduğumuz veri setinde de denenmiştir. Kaynak [9] için kendi veri setimizle elde ettiğimiz sonuç %71,7'dir ve bu değer yazarların 70,8'lik başarımlarına oldukça yakındır. Önerilen yöntemin hesaplama süresini değerlendirmek için 2000×3000 ölçekli sekiz adet görüntü üzerinde deneyler yapılmış ve sonuçlar Tablo 4.4'te gösterilmiştir. Tablo'da da görüldüğü üzere geliştirdiğimiz DPA yönteminin, en iyi ikinci başarıma sahip olan Zhu ve arkadaşlarının [10] çalışmasından daha hızlı olduğu gözlenmiştir.

Tablo 4.4. Önerilen yöntemin işlem süresi bakımından karşılaştırılması

Havaalanları	Süre (sn)	
	Geliştirilmiş DPA	Zhu ve ark. [10]
Adana (Türkiye)	9.41	13.73
Adıyaman (Türkiye)	8.45	12.57
Ağrı (Türkiye)	5.01	8.68
Bandırma (Türkiye)	4.95	9.09
Bari (İtalya)	6.23	10.59
Bingöl (Türkiye)	4.39	7.69
Bologna (İtalya)	11.85	15.84
Cuneo (İtalya)	6.31	9.63
Ortalama süre	7.08	10.98

Önerilen ikinci yöntemi test etmek için de elde etmiş olduğumuz aynı veri seti kullanılmıştır. Önerilen birinci yöntemde olduğu gibi 92 veri setinin 48'i eğitim ve 44'ü de test için kullanılmış ve eğitim setindeki her aday bölge 224×224×3 ölçeğine ayarlanmıştır. Geliştirilen DPA'nın eğitim veri setine uygulanması sonucu elde edilen ve uygun ölçeğe dönüştürülen havaalanı ve havaalanı olmayan aday bölgelere ait bazı örnekler Şekil 4.2 ve Şekil 4.3'te gösterilmiştir.

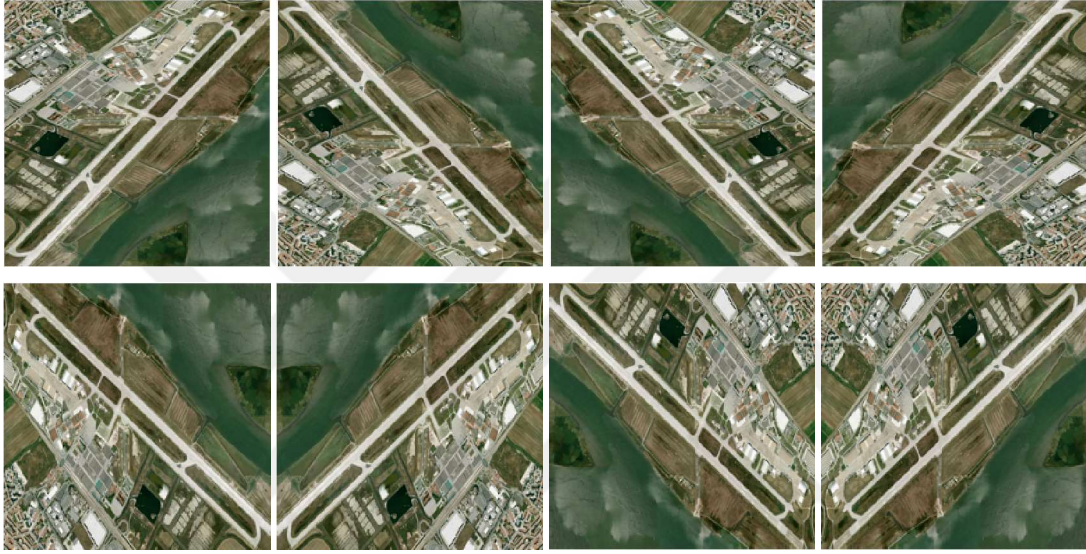


Şekil 4.2. Havaalanı bölgeleri



Şekil 4.3. Havaalanı olmayan bölgeler

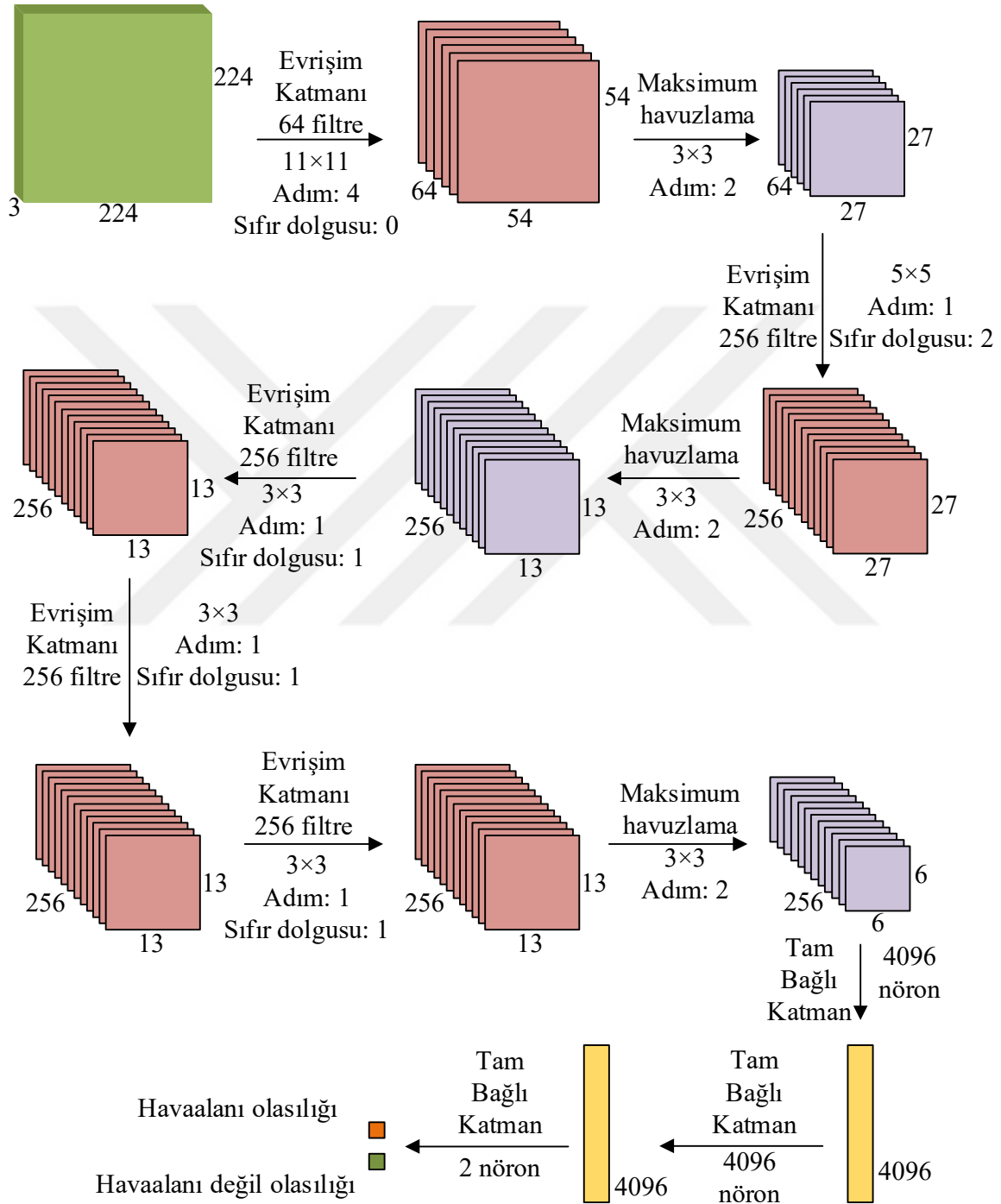
Bu aday görüntüler bir önceki yöntemde elde edilen aday bölgelerle aynı görüntülerdir. Sadece görüntü ölçeği normalleştirilmiştir. Bununla birlikte aşırı uyumu azaltmak için eğitim veri setinde tespit edilen aday bölgeler yapay olarak artırılmıştır. Öncelikle girdi görüntüleri sırasıyla X, Y ve XY eksenine göre çevrilmiştir. Daha sonra aynı girdi görüntüleri 90° ve -90° döndürülerek Y eksenine göre çevrilmiştir. Dolayısıyla eğitim veri seti yapay olarak $8 \times (67+271)=2704$ adete çıkarılmıştır. Şekil 4.4'te X, Y ve XY eksenlerinde ters çevrilen bir veri artırma örneği gösterilmiştir.



Şekil 4.4. Veri artırma işlemi. Birinci satır sırasıyla: orijinal görüntü, X, Y ve XY eksenine göre çevrilmiş görüntü. İkinci satır sırasıyla: 90° döndürülmüş, Y eksenine çevrilmiş, -90° döndürülmüş ve Y eksenine çevrilmiş görüntü

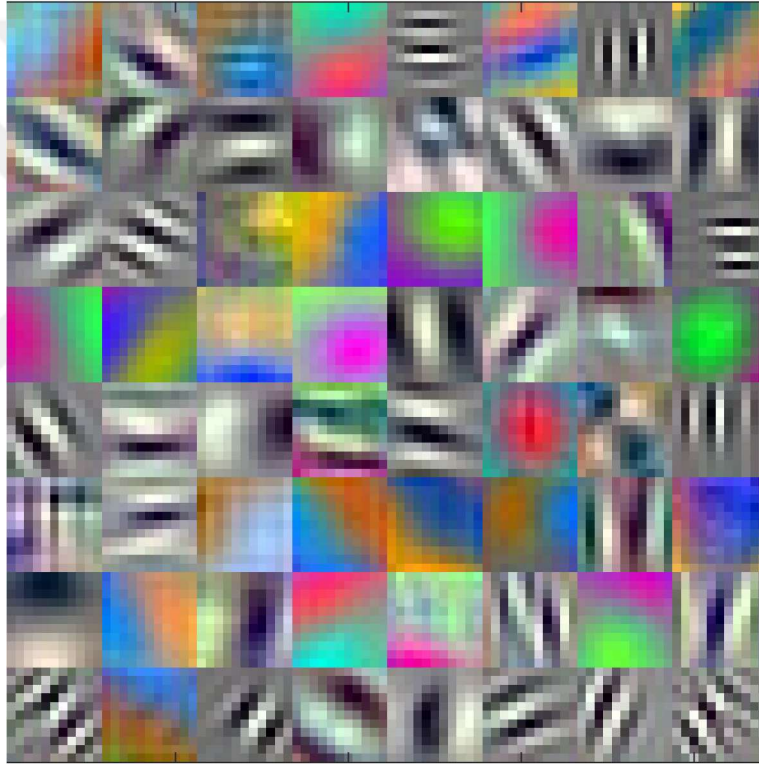
ESA'ya yapılan girdi $224 \times 224 \times 3$ boyutlu renkli görüntüdür. ESA modeli beş evrişim ve üç tam bağlı katmana sahiptir. Birinci evrişim katmanında 11×11 'lik 64 adet filtre kullanılmıştır. Evrişim adım sayısı 4 pikseldir. Düzeltme doğrusal birim ve yerel yanıt normalleştirme katmanı birinci ve ikinci evrişim katmanlarını takip etmektedir. Bazı evrişim katmanlarını takip eden mimaride beş adet maksimum havuzlama katmanı bulunmaktadır. Havuzlama işlemi iki adımlı 3×3 piksellik bir pencere üzerinden gerçekleştirmiştir. İkinci evrişim katmanı, önceki katmanın çıktısını 256 adet 5×5 boyutunda filtre ile filtrelemiştir. Evrişim adımı bir piksel ve uzaysal dolgu iki pikseldir. Üçüncü evrişim katmanında ise 3×3 boyutlu 256 adet filtre kullanılmıştır. Evrişim adımı ve uzaysal dolgu bir pikseldir. Üçüncü evrişim katmanını bir düzeltme doğrusal birim katmanı takip etmektedir. Dördüncü ve beşinci evrişim katmanları üçüncü evrişim katmanı ile aynı yapıya sahiptir. Daha önce belirtildiği gibi üç tam bağlı katman, evrişim katmanlarını takip etmektedir. Her tam bağlı

katmanın 4096 düğümü vardır. Bırakma (dropout) olasılığı 0,5 olarak seçilmiştir. Son olarak son katman için bir kayıp katmanı kullanılmıştır. Kullanılan ESA mimarisi Şekil 4.5'te gösterilmiştir.



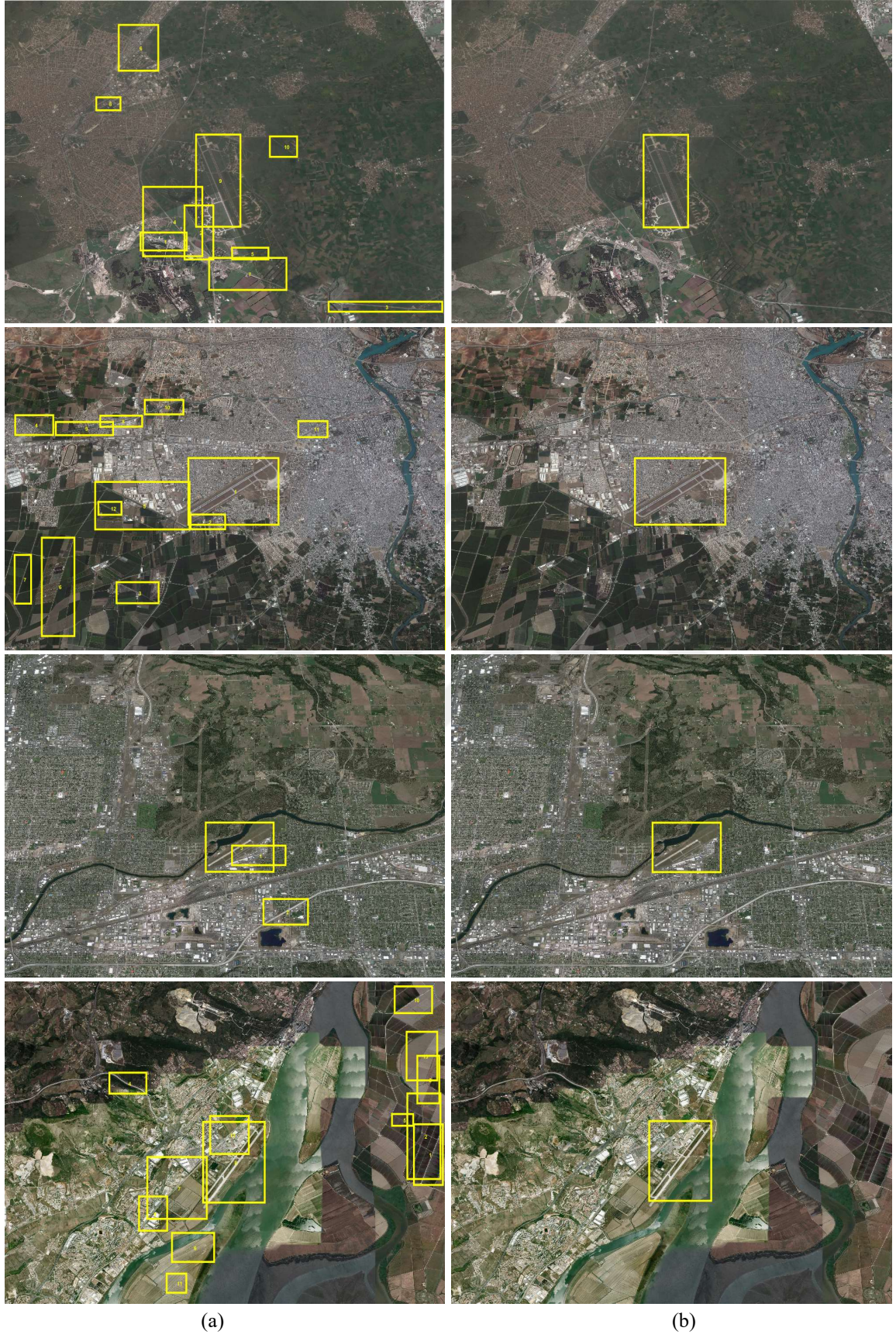
Şekil 4.5. Havaalanı tespiti için kullanılan ESA mimarisi

ESA modelini üretmek için bir olasılıksal (stokastik) gradyan düşmeli MaTConvNet aracı kullanılmıştır. Eğitim süresince 70 eğitim örneklerinin toplu boyutu dikkate alınmıştır. Öğrenme oranı, logaritmik alanda uygun aşamalı bir adımla değiştirilmiştir. Kayıp katmanda softmax fonksiyonu kullanılmıştır. Her katmandaki ağırlıkların başlangıcı sıfır '0' ortalama ve 0,01 standart sapmaya sahip bir Gauss dağılımı üzerinden ele alınmıştır. Tam bağlı gizli katmanlarda olduğu gibi sinir sapmaları ikinci, dördüncü ve beşinci evrişim katmanlarında bir (1) değerine başlatılmıştır. Şekil 4.6'da $224 \times 224 \times 3$ 'lük giriş görüntüsünde birinci evrişim katmanı tarafından öğrenilen $11 \times 11 \times 3$ boyutunda 64 adet evrişim çekirdeği gösterilmiştir.



Şekil 4.6. Birinci evrişim katmanı tarafından öğrenilen $11 \times 11 \times 3$ boyutunda 64 adet evrişim çekirdeği

Şekil 4.7'de başarılı bir şekilde tespit edilmiş havaalanı görüntüleri gösterilmiştir. Tespit edilen havaalanı aday görüntüleri (a) sütununda, doğrulanmış havaalanı görüntüleri de (b) sütununda belirtilmiştir.



Şekil 4.7. Başarılı bir şekilde tespit edilmiş havaalanı görüntüleri; (a) tespit edilmiş havaalanı aday görüntüleri, (b) doğrulanmış havaalanları

Önerilen ikinci yöntemin test setinde genel performansını değerlendirmek için doğruluk, duyarlılık ve özgüllük ölçütleri kullanılmıştır. Elde edilen sonuçlar Tablo 4.5'te gösterilmiştir.

Tablo 4.5. Önerilen ikinci yöntemin performansı

Sınıflandırıcı yapısı	Doğruluk	Duyarlılık	Özgüllük
ESA	%95,21	%94,12	%95,51

Tablo 4.5'te görüldüğü gibi önerilen ESA modeli ile %95,21 doğruluk, %94,12 duyarlılık ve %95,51 özgüllük değeri elde edilmiştir. Test görüntüleri için 223 aday bölge sınıflandırılırken sadece 10 tane aday bölge yanlış olarak sınıflandırılmıştır. Bu sonuçlar ESA modelinin hem havaalanı hem de havaalanı olmayan bölgeleri başarıyla sınıflandırdığını göstermektedir.

5. SONUÇLAR

Son yıllarda uydu görüntülerinden otomatik hedef tespiti önemli bir uygulama olarak karşımıza çıkmaktadır. Yüksek çözünürlüklü UAG'ler yollar, binalar, orman arazileri, araçlar gibi hedeflerin tespitinde sıkça kullanılmaktadır. Hem askeri hem de sivil uygulamalar için vazgeçilmez alanlar olarak düşünülen havaalanları da yine bu uygulamalar içinde önemli bir yer teşkil etmektedirler. Ancak havaalanlarının zeminleri karmaşık olma eğilimindedirler ve genelde yollar, nehirler ve binaların bulunduğu şehir içi ya da şehirlerarasındaki bölgeler, havaalanı ile benzer özellikler gösterebilmektedir. Ayrıca hedef tespit sistemlerinde kullanılan uydu görüntüleri çoğunlukla yüksek çözünürlüğe sahiptir. Bu nedenle kullanılan algoritmaların düşük işlem karmaşıklığına sahip olması ve tespit işlemini daha kısa sürede gerçekleştirebilmesi arzu edilmektedir.

Bu tez çalışmasında havaalanı tespit problemleri için iki farklı yöntem önerilmiştir. Birinci yöntemde havaalanı tespiti için geliştirilmiş DPA, ÖBÖD ve FV gösterimine dayalı yeni bir yöntem önerilmiştir. Aday havaalanı bölgelerinin çıkarılması için yeni bir doğru parçasına dayalı prosedür kullanılmıştır. Bu da geliştirmiş olduğumuz DPA yöntemi ile elde edilmiştir. Bu aday bölgeler için ÖBÖD öznelikleri ve FV betimleyicisi uygulanmıştır. Doğrulama aşamasında ise DVM tabanlı sınıflandırıcı kullanılmıştır.

DeneySEL sonuçlar, homojen haritalama ve LIBLINEAR DVM yapısının birlikte kullanıldığı durumda en iyi doğruluğun elde edildiğini ortaya koymaktadır. Sınıflandırma doğruluğu %94,6'dır. Ayrıca duyarlılık ve özgüllük değerleri sırasıyla %78 ve %100'dür. En kötü sınıflandırma başarımları ise, %82,5'lik doğruluk, %56'lık duyarlılık ve %90,2'lik özgüllük değerleri ile doğrusal çekirdek yapısının kullanıldığı DVM olmuştur. Bu nedenle LIBLINEAR yapısının sınıflandırma performansını açık bir şekilde artırdığı anlaşılmaktadır. LIBLINEAR DVM yapısı için sonuçlar DVM'den yaklaşık %10 daha yüksektir. Ayrıca LIBLINEAR DVM yapısı üzerinde homojen haritalama ile başarımlar sonucunu yaklaşık %2 daha artırılmıştır.

DeneySEL çalışmalar önerilen birinci yöntemin literatürdeki diğer yöntemlerden daha iyi performansa sahip olduğunu göstermiştir. Ayrıca yöntem ikinci en iyi yöntem olan Kaynak [10]'dan %6 daha doğru olmasının yanında %35,5 daha hızlıdır.

Önerilen ikinci yöntemde ise havaalanı tespit problemleri için derin ESA kullanımı araştırılmıştır. Potansiyel havaalanı aday bölgeleri yine geliştirmiş olduğumuz DPA ile tespit

edildikten sonra havaalanı tanıma amacıyla bir ESA mimarisi eğitilmiştir. Kullanılan ESA mimarisi beş evrişim, üç havuzlama ve üç tam bağlı katmandan oluşmaktadır. ESA modeli, havaalanlarını karakterize eden belirgin özellikleri çıkarmanın yanında diğer yöntemlerden daha iyi performans gösteren bir sınıflandırma da gerçekleştirmektedir. Yöntemin test setinde genel performansını değerlendirmek için doğruluk, duyarlılık ve özgüllük ölçütleri kullanılmıştır. ESA modeli ile %95,21 doğruluk, %94,12 duyarlılık ve %95,51 özgüllük değeri elde edilmiştir. Test görüntüleri için 223 aday bölge sınıflandırılırken sadece 10 tane aday bölge yanlış olarak sınıflandırılmıştır. Elde edilen sonuçlar ESA modelinin hem havaalanı hem de havaalanı olmayan bölgeleri başarıyla sınıflandırdığını göstermektedir. Ayrıca ESA modelinin kullanıldığı ikinci yöntem, önerilen birinci yöntemden yaklaşık %1’lik bir başarımla üstünlüğü sağlamıştır.

Sonuç olarak bu tez çalışmasında havaalanı tespiti problemleri için iki özgün yöntem önerilmiştir. Deneysel çalışmalar MATLAB ortamında tasarlanmıştır. Elde edilen sonuçlara göre önerdiğimiz iki yöntem de literatürdeki en iyi başarımla sonucuna sahiptir.

Gelecek süreçte UAG’lerden havaalanı tespitinin başarımlarını daha da artıracak farklı yöntemler üzerinde araştırmalar yapılmaya devam edilecektir.

KAYNAKLAR

- [1] **Aytekin, Ö., Zöngür, U., and Halici, U.**, 2013. Texture-based airport runway detection, *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 3, pp. 471–475.
- [2] **Wang, W., Liu, L., Hu, C., Jiang, Y., and Kuang, G.**, 2011. Airport detection in SAR image based on perceptual organization. In *Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011 International Workshop on IEEE*, pp. 1-5.
- [3] **Wang, X., Wang, B., and Zhang, L.**, 2011. Airport detection in remote sensing images based on visual attention. In *International Conference on Neural Information Processing*, Springer, Berlin, Heidelberg, pp. 475-484.
- [4] **Yang, S. L., Lu, L., Du, Z., Liang, J., and Li, S.**, 2006. Automatic recognition of airport runways in SAR images. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, vol 30, no. 1, pp. 56-59.
- [5] **Tang, D. J., Zhou, X. M., Zhao, G. Q., and Li, E. S.**, 2010. Improved phase group baseline algorithm application in the airfield extraction based on rs image. *Image Technology*, vol 1, 39-43.
- [6] **Wang, Z. L., and Wu, L. H.**, 2008. A fast Method of Airport Runway Detection Based on Mathematical Morphology [J]. *Computer Technology and Development*, vol 18, no. 7, 193-196.
- [7] **Kou, Z., Shi, Z., and Liu, L.**, 2012. Airport detection based on line segment detector. In *Computer Vision in Remote Sensing (CVRS), 2012 International Conference on IEEE*, pp. 72-77.
- [8] **Wang, X., Lv, Q., Wang, B., & Zhang, L.**, 2013. Airport detection in remote sensing images: a method based on saliency map. *Cognitive neurodynamics*, 7(2), April pp. 143-154.
- [9] **Qu, Y., Li, C., and Zheng, N.**, 2005. Airport detection base on support vector machine from a single image. In *Information, Communications and Signal Processing, Fifth International Conference on IEEE*, pp. 546-549.

- [10] **Zhu, D., Wang, B., and Zhang, L.,** 2015. Airport target detection in remote sensing images: A new method based on two-way saliency. *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1096-1100.
- [11] **Von Gioi, R. G., Jakubowicz, J., Morel, J. M., and Randall, G.,** 2010. LSD: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722-732.
- [12] **Harel, J., Koch, C., and Perona, P.,** 2007. Graph-based visual saliency. In *Advances in neural information processing systems*, pp. 545-552.
- [13] **Tao, C., Tan, Y., Cai, H., and Tian, J.,** 2011. Airport detection from large IKONOS images using clustered SIFT keypoints and region information. *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 128-132.
- [14] **Zuo, Z., Chen, H., and Zhang, T.,** 2009. Study on hit-aim detection of airfield runway based on weighted structure templates matching. In *Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*, International Society for Optics and Photonics, pp. 749532-749532.
- [15] **Wu, W., Xia, R., Xiang, W., Hui, B., Chang, Z., Liu, Y., and Zhang, Y.,** 2014. Recognition of airport runways in FLIR images based on knowledge. *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 9, pp. 1534-1538.
- [16] **Zhu, D., Wang, B., and Zhang, L.,** 2014. Two-way saliency for airport detection in remote sensing images. In *Audio, Language and Image Processing (ICALIP), 2014 International Conference on IEEE*, July, pp. 526-531.
- [17] **Dong, Y., Yuan, B., Wang, H., and Shi, Z.,** 2011. A runway recognition algorithm based on heuristic line extraction. In *Image Analysis and Signal Processing (IASP), 2011 International Conference on IEEE*, October, pp. 292-296.
- [18] **Wang, M., and Zhang, X.,** 2007. An effective method to detect straight lines from high spatial resolution remotely sensed imagery and its applications for runway extraction. In *Geoinformatics 2007: Remotely Sensed Data and Information*, International Society for Optics and Photonics, August, vol. 6752, p. 67520U.

- [19] **Tripathi, A. K., and Swarup, S.,** 2013. Shape and color features based airport runway detection. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International IEEE*, February, pp. 836-841.
- [20] **Jinglin, Y., Linbo, T., Baojun, Z., Chenwei, D., and Baoxian, W.,** 2013. Airport target detection algorithm in remote sensing images based on JPEG2000 compressed domain. In *IET International Radar Conference*, pp. 0491-0491.
- [21] **Tang, G., Xiao, Z., Liu, Q., and Liu, H.,** 2015. A Novel Airport Detection Method via Line Segment Classification and Texture Classification. *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2408-2412.
- [22] **Voorhees, H., and Poggio, T.,** 1987. Detecting textons and texture boundaries in natural images. In *Proceedings of the First International Conference on Computer Vision*, June, vol. 59, pp. 250-258.
- [23] **Juneja, M., and Sandhu, P. S.,** 2009. Performance evaluation of edge detection techniques for images in spatial domain. *International journal of computer theory and Engineering*, vol. 1, no. 5, December, pp. 1793-8201.
- [24] **Pratt, William K.,** 2007. Digital image processing. 4 th edition, John WileyInc.
- [25] **Marr, D., and Hildreth, E.,** 1980. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167), pp. 187-217.
- [26] **Dellepiane, S., Giusto, D. D., Serpico, S. B., and Vernazza, G.,** 1989. Automatic parameter computation for edge detection by the zero-crossing method. In *12° Colloque sur le traitement du signal et des images, FRA, GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*, 12-16 juin, pp. 617-620.
- [27] **Haralick, R. M., and Lee, J. S.,** 1988. Context dependent edge detection. In *Pattern Recognition, 9th International Conference on IEEE*, November, pp. 203-207.
- [28] **Canny, J.,** 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), pp. 679-698.
- [29] **Deriche, R.,** 1987. Optimal edge detection using recursive filtering. *International Journal of Computer Vision*, 2, London, (June 8-12), pp. 167-187.
- [30] **Deriche, R.,** 1990. Fast algorithms for low-level vision. *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 1, pp. 78-87.

- [31] **Shen, J., and Castan S.**, 1990. Edge detection by sign correspondence for zero-crossings. Actes du Premier Colloque Image' COIn, Bordeaux, 19-21 November, pp. 279-284.
- [32] **Shen, J., and Castan, S.**, 1992. An optimal linear operator for step edge detection. *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 2, pp. 112-133.
- [33] **Leclerc, Y. G., and Zucker, S. W.**, 1987. The local structure of image discontinuities in one dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3), pp. 341-355.
- [34] **Matthews, J.**, (2002). An introduction to edge detection: The sobel edge detector. Available at <http://www.generation5.org/content/2002/im01.asp>.
- [35] **Roberts, L. G.**, 1965. Machine perception of 3-D solids. *Optical and Electro-Optical Information Processing*. MIT Press.
- [36] **Gonzalez, R. C., and Woods R. E.**, 2002. Digital Image Processing. 2nd ed. Prentice Hall.
- [37] **Torre, V., and Poggio, T. A.**, 1986. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, March, pp. 147-163.
- [38] **Grimson, W. E. L., and Hildreth, E. C.**, 1985. Comments on Digital step edges from zero crossings of second directional derivatives. *IEEE transactions on pattern analysis and machine intelligence*, (1), pp. 121-127.
- [39] **Haralick, R. M.**, 1984. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 1, pp. 58-68.
- [40] **Canny, J. F.**, 1983. Finding edges and lines in images. Master's thesis, MIT.
- [41] **Burns, J. B., Hanson, A. R., and Riseman, E. M.**, 1986. Extracting straight lines. *IEEE transactions on pattern analysis and machine intelligence*, no. 4, pp. 425-455.
- [42] **Desolneux, A., Moisan, L., and Morel, J. M.**, 2000. Meaningful alignments. *International Journal of Computer Vision*, vol. 40, no. 1, pp. 7-23.

- [43] **Desolneux, A., Moisan, L., and Morel, J. M.**, 2007. From gestalt theory to image analysis: a probabilistic approach. vol. 34, *Springer Science & Business Media*.
- [44] **Lowe, D. G.**, 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, vol. 60, no. 2, pp. 91-110.
- [45] **Witkin, A.P.**, 1983. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 1019-1022.
- [46] **Koenderink, J. J.**, 1984. The structure of images. *Biological cybernetics*, vol. 50, no. 5, pp. 363-370.
- [47] **Lindeberg, T.**, 1994. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, vol. 21, no. 2, pp. 224-270.
- [48] **Lowe, D. G.**, 1999. Object recognition from local scale-invariant features. In *Computer vision, The proceedings of the seventh IEEE international conference on IEEE*, vol. 2, pp. 1150-1157.
- [49] **Mikolajczyk, K., and Schmid, C.**, 2002. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, pp. 128-142.
- [50] **Brown, M., and Lowe, D. G.**, 2002. Invariant Features from Interest Point Groups. In *British Machine Vision Conferenc (BMVC)*, September, Cardiff, Wales, vol. 4, pp. 656–665.
- [51] **Harris, C., and Stephens, M.**, 1988. A combined corner and edge detector. In *Fourth Alvey vision conference*, August, Manchester, UK, vol. 15, no. 50, pp. 147-151.
- [52] **Schmid, C., and Mohr, R.**, 1997. Local grayvalue invariants for image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 5, pp. 530-535.
- [53] **Jaakkola, T., and Haussler, D.**, 1999. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems (NIPS)*, pp. 487-493.
- [54] **Döngel, T.**, 2016. Semi-Supervised Visual Representations for Large Scale Image Databases. March, Master's of thesis.

- [55] **Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J.,** 2013. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, vol. 105, no. 3, pp. 222-245.
- [56] **Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H.,** 2010. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), Conference on IEEE*, June, pp. 3384-3391.
- [57] **Titterton, D. M., Smith A. F. M., and Makov, U. E.,** 1985. Statistical Analysis of Finite Mixture Distributions. John Wiley, New York.
- [58] **Perronnin, F., Dance, C., Csurka, G., and Bressan, M.,** 2006. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision (ECCV)*, Springer, Berlin, Heidelberg, May, pp. 464-475.
- [59] **Winn, J., Criminisi, A., and Minka, T.,** 2005. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on IEEE*, October, vol. 2, pp. 1800-1807.
- [60] **Krapac, J., Verbeek, J., and Jurie, F.,** 2011. Modeling spatial layout with fisher vectors for image categorization. In *Computer Vision (ICCV), International Conference on IEEE*, November, pp. 1487-1494.
- [61] **Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B.,** 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, vol 13, no. 4, pp. 18-28.
- [62] **Krizhevsky, A., Sutskever, I., and Hinton, G. E.,** 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097-1105.
- [63] **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., and Rabinovich, A.,** 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, arXiv preprint arXiv:1409.4842.
- [64] **Yi, D., Lei, Z., and Li, S. Z.** 2014. Age estimation by multi-scale convolutional network. In *Asian Conference on Computer Vision (ACCV)*, November, Springer, Cham, pp. 144-158.

- [65] **Kavukcuoglu, K., Sermanet, P., Boureau, Y. L., Gregor, K., Mathieu, M., and Cun, Y. L.,** 2010. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pp. 1090-1098.
- [66] **Rautkorpi, R., and Iivarinen, J.,** 2004. A novel shape feature for image classification and retrieval. *Image Analysis and Recognition*, Springer, pp. 753-760.
- [67] **Zhu, S. C., Zhang, R., and Tu, Z.,** 2000. Integrating bottom-up/top-down for object recognition by data driven Markov chain Monte Carlo. In *Computer Vision and Pattern Recognition, Proceedings. IEEE Conference on*, vol. 1, pp. 738-745.
- [68] **Hubel, D. H., & Wiesel, T. N.,** 1968. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, vol 195, no. 1, pp. 215-243.
- [69] **Fukushima, K., and Miyake, S.,** 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, Springer, Berlin, Heidelberg, pp. 267-285.
- [70] **LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D.,** 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, vol 1, no. 4, pp. 541-551.
- [71] **LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.,** 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.
- [72] **Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.,** 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of machine learning research*, vo. 15, no. 1, pp. 1929-1958.
- [73] **Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V.,** 2014. Neural codes for image retrieval. In *European conference on computer vision*, September, Springer, Cham, pp. 584-599.

- [74] **Korekado, K., Morie, T., Nomura, O., Ando, H., Nakano, T., Matsugu, M., and Iwata, A.,** 2003. A convolutional neural network VLSI for image recognition using merged/mixed analog-digital architecture. In *Knowledge-Based Intelligent Information and Engineering Systems*, Springer, Berlin, Heidelberg, pp. 169-176.
- [75] **Abdel-Hamid, O., Deng, L., and Yu, D.,** 2013. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech*, August, pp. 3366-3370.
- [76] **Scherer, D., Müller, A., and Behnke, S.,** 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *Artificial Neural Networks–ICANN 2010*, pp. 92-101.
- [77] **Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., and Gambardella, L. M.,** 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), IEEE International Conference on*, November, pp. 342-347.
- [78] **Boureau, Y. L., Ponce, J., and LeCun, Y.,** 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 111-118.
- [79] **Xu, B., Wang, N., Chen, T., and Li, M.,** 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.
- [80] **Simonyan, K., and Zisserman, A.,** 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [81] **Nowlan, S. J., and Hinton, G. E.,** 1992. Simplifying neural networks by soft weight-sharing. *Neural computation*, vol. 4, no. 4, pp. 473-493.
- [82] **Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J.,** 2011. Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, July, vol. 22, no. 1, p. 1237.

- [83] **Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. A.,** 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, vol. 11, December, pp. 3371-3408.
- [84] **Ba, J., and Frey, B.,** 2013. Adaptive dropout for training deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 3084-3092.
- [85] **Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C.,** 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* October, Citeseer, vol. 1631, p. 1642.
- [86] **Powers, D. M.,** 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learning Technology*, 2:1, pp. 37-63.
- [87] **He, H., and Garcia, E. A.,** 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, vol 21, no. 9, pp. 1263-1284.
- [88] **Bengio, Y., and Grandvalet, Y.,** 2004. No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, 5(Sep), pp. 1089-1105.
- [89] **Kohavi, R.,** 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, vol. 14, no. 2, pp. 1137-1145.
- [90] **Google Earth Pro,** Uydudan Havaalanı Görüntüleri Temini, Son erişim: 10 Mart 2015.
- [91] <http://www.vlfeat.org/> Açık kaynak kodlu bilgisayarlı görme algoritmaları kütüphanesi. 20 Mayıs 2014.
- [92] **Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J.,** 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research*, 9(Aug), pp. 1871-1874.

EKLER

EK-1: Veri setinde kullanılan havaalanı ve görüntü bilgileri

Havaalanı	Şehir	Ülke	Enlem	Boylam	Yükseklik (piksel)	Genişlik (Piksel)
Alverca	Vila Franca de Xira	Portekiz	38.8833008	-9.0300999	2000	3000
Bagotville	Bagotville	Canada	48.3306007	-70.9963989	2000	3000
Bisho	Bisho	South Africa	-32.8970985	27.2791004	2000	3000
Brandon Municipal	Brandon	Canada	49.9099998	-99.9518967	2000	3000
Charlottetown	Charlottetown	Canada	46.2900009	-63.1211014	2000	3000
Coronel Fap Francisco Secada Vignetta	Iquitos	Peru	-3.78474	-73.3087997	2000	3000
Cuidad	Cuidad	İspanya	38.867179	-3.992693	2000	3000
Eloy Municipal	Eloy	United States	32.8069992	-111.586998	2000	3000
Felts Field	Spokane	United States	47.6830556	-117.3225	2000	3000
Francazal	Toulouse/Francazal	France	43.5456009	1.3674999	2000	3000

General Mitchell	Milwaukee	United States	42.9472008	-87.8965988	2000	3000
Phifer Field	Wheatland	United States	42.0583333	-104.9194444	2000	3000
General Jose Antonio	Venezuela	Barcelona	10.1070995	-64.6892014	2000	3000
Kansas City	Kansas City	United States	39.2975998	-94.7138977	2000	3000
Kasteli	Crete Island	Greece	35.1920013	25.3269997	2000	3000
King Salmon	King Salmon	United States	58.6767998	-156.6490021	2000	3000
Sydney Kingsford Smith	Sydney	Australia	-33.9460983	151.177002	2000	3000
Košice	Košice	Slovakia	48.6631012	21.2411003	2000	3000
Lechfeld	Lechfeld	Almanya	48.1855011	10.8612003	2000	3000
Leon	Leon	Spain	42.590370	-5.643263	2000	3000
Macdill Afb	Avon Park	United States	27.6506004	-81.3494034	2000	3000
Makung	Makung	Taiwan	23.5687008	119.6279984	2000	3000
Meridian Nas	Meridian	United States	32.555141	-88.559335	2000	3000
Montgomery	Conroe	United States	32.303474	-86.391526	2000	3000
Octeville	Le Havre	France	49.5339012	0.088056	2000	3000

Page Field	Fort Myers	United States	26.5833333	-81.8666667	2000	3000
Point Mugu Nas	Point Mugu	United States	34.1202778	-119.1211111	2000	3000
Port Elizabeth	Port Elizabeth	South Africa	-33.9849014	25.6173	2000	3000
Porto Nacional	Porto Nacional	Brazil	-10.7194004	-48.3997002	2000	3000
Quetzalcóatl	Nuevo Laredo	Mexico	27.4438992	-99.5705032	2000	3000
Rafael Cabrera	Nueva Gerona	Küba	21.837694	-82.780378	2000	3000
Ramstein	Ramstein	Germany	49.4369011	7.6002798	2000	3000
Saskatoon John G. Diefenbaker	Saskatoon	Canada	52.1707993	-106.6999969	2000	3000
Schleswig	Jagel	Germany	54.459301	9.5163298	2000	3000
Sherman AAF	Fort Leavenworth	United States	39.3675	-94.9144444	2000	3000
Shreveport Regional	Shreveport	United States	32.4466019	-93.8255997	2000	3000
Sitka Rocky Gutierrez	Sitka	United States	57.0471001	-135.3619995	2000	3000
Spokane International	Spokane	United States	47.6198997	-117.5339966	2000	3000
M. R. Štefánik	Bratislava	Slovakia	48.1702003	17.2126999	2000	3000
Takhli	Nakhon Sawan	Thailand	15.2772999	100.2959976	2000	3000

Terre Haute International Hulman Field	Terre Haute	United States	39.4514999	-87.3076019	2000	3000
Teterboro	Teterboro	United States	40.85	-74.0608333	2000	3000
Suffolk County	Westhampton Beach	United States	40.8333333	-72.6166667	2000	3000
Townsville	Townsville	Australia	-19.2525005	146.7649994	2000	3000
Pierre Van Ryneveld	Upington	South Africa	-28.3990993	21.2602005	2000	3000
Whiteman AFB	Knob Noster	United States	38.7302778	-93.5480556	2000	3000
Wilmington International	Wilmington	United States	34.2705994	-77.9026031	2000	3000
Zurich	Zurich	Switzerland	47.4646988	8.5491695	2000	3000
Adana	Adana	Türkiye	36.9822006	35.2803993	2000	3000
Adıyaman	Adıyaman	Türkiye	37.7313995	38.4688988	2000	3000
Ağrı	Ağrı	Türkiye	39.654541	43.0259781	2000	3000
Amasya Merzifon	Amasya	Türkiye	40.8293991	35.5219994	2000	3000
Esenboğa	Ankara	Türkiye	40.1281013	32.9950981	2000	3000
Antalya	Antalya	Türkiye	36.8987007	30.8005009	2000	3000
Gazipaşa	Antalya	Türkiye	36.2992172	32.3005981	2000	3000

Balıkesir Merkez	Balıkesir	Türkiye	39.6193008	27.9260006	2000	3000
Balıkesir Körfez	Balıkesir	Türkiye	39.5545998	27.0137997	2000	3000
Bari	Bari	İtalya	41.1389008	16.7605991	2000	3000
Bingöl	Bingöl	Türkiye	38.860838	40.589485	2000	3000
Bologna / Borgo Panigale	Bologna	İtalya	44.5354004	11.2887001	2000	3000
Cuneo / Levaldigi	Cuneo	İtalya	44.5470009	7.62322	2000	3000
Dalaman	Muğla	Türkiye	36.7131004	28.7924995	2000	3000
Çardak	Denizli	Türkiye	37.7855988	29.7012997	2000	3000
Diyarbakır	Diyarbakır	Türkiye	37.898483	40.196791	2000	3000
Elazığ	Elazığ	Türkiye	38.6068993	39.2914009	2000	3000
Erzincan	Erzincan	Türkiye	39.7102013	39.5270004	2000	3000
Erzurum	Erzurum	Türkiye	39.956501	41.1702003	2000	3000
Anadolu	Eskişehir	Türkiye	39.8098984	30.5193996	2000	3000
Gaziantep	Gaziantep	Türkiye	36.9472008	37.4786987	2000	3000
İncirlik	Adana	Türkiye	37.0021019	35.4258995	2000	3000

Süleyman Demirel	Isparta	Türkiye	37.8554001	30.3684006	2000	3000
Atatürk	İstanbul	Türkiye	40.9768982	28.8146	2000	3000
Sabiha Gökçen	İstanbul	Türkiye	40.8986015	29.3092003	2000	3000
Adnan Menderes	İzmir	Türkiye	38.2924004	27.1569996	2000	3000
Kahramanmaraş	Kahramanmaraş	Türkiye	37.538826	36.9535217	2000	3000
Erkilet	Kayseri	Türkiye	38.770401	35.4953995	2000	3000
Lamezia Terme	Lamezia Terme	İtalya	38.9053993	16.2423	2000	3000
Erhaç	Malatya	Türkiye	38.4352989	38.0909996	2000	3000
Muş	Muş	Türkiye	38.7477989	41.6612015	2000	3000
Nápoli	Nápoli	İtalya	40.8860016	14.2908001	2000	3000
Kapadokya	Nevşehir	Türkiye	38.7719002	34.5345001	2000	3000
Olbia	Olbia	İtalya	40.8987007	9.5176296	2000	3000
Pisa	Pisa	İtalya	43.6838989	10.3927002	2000	3000
Çarşamba	Samsun	Türkiye	41.2545013	36.5671005	2000	3000
Çorlu	Tekirdağ	Türkiye	41.1381989	27.9190998	2000	3000

Tokat	Tokat	Türkiye	40.3074303	36.3674088	2000	3000
Trabzon	Trabzon	Türkiye	40.9950981	39.7896996	2000	3000
Trieste	Trieste	İtalya	45.8274994	13.4722004	2000	3000
Şanlıurfa GAP	Şanlıurfa	Türkiye	37.4456635	38.8955917	2000	3000
Uşak	Uşak	Türkiye	38.6814995	29.4717007	2000	3000
Van Ferit Melen	Van	Türkiye	38.4682007	43.3322983	2000	3000
Zonguldak	Zonguldak	Türkiye	41.5064011	32.0886002	2000	3000

ÖZGEÇMİŞ

1985 yılında İzmir’de doğdum. İlk, orta ve lise öğrenimimi İzmir’de tamamladım. 2005 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik Öğretmenliği Bölümü’nde lisans öğrenimime başladım. 2011 yılında “Çok Seviyeli Eviricili Elektronik Balast Devresinin Modellenmesi ve Benzetimi” başlıklı yüksek lisans tezini tamamladım. 2013-2014 yılı bahar dönemi itibariyle doktora başladım. Halen Bitlis Eren Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik - Elektronik Mühendisliği Bölümü’nde Araştırma Görevlisi olarak çalışmaktayım.

