

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İMALAT SİSTEMLERİNDE DERİN ÖĞRENME TABANLI DOKU HATA
TESPİTİ**



DOKTORA TEZİ

Hüseyin ÜZEN

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Davut HANBAY

Ekim 2022

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İMALAT SİSTEMLERİNDE DERİN ÖĞRENME TABANLI DOKU HATA
TESPİTİ**

DOKTORA TEZİ

**Hüseyin ÜZEN
(36183619045)**

Bilgisayar Mühendisliği Anabilim Dalı

**Tez Danışmanı: Prof. Dr. Davut HANBAY
Eş Danışman: Doç. Dr. Muammer TÜRKÖĞLU**

Ekim 2022

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının her aşamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgmeden beni her konuda yönlendiren danışman hocam Sayın Prof. Dr. Davut HANBAY'a,

Akademik eğitimim boyunca bana destek olan ve tez yazımım süresince bilgi ve birikimlerini benden esirgemeyen kıymetli hocalarım Prof. Dr. Berrin A. YANIKOĞLU, Doç. Dr. Muammer TÜRKÖĞLU, Dr. Öğr. Üyesi Ali ARI, Dr. Öğr. Üyesi Muzaffer ASLAN ve Dr. Öğr. Üyesi Dursun ÖZTÜRK'e

Görev yaptığım üniversitede bulunduğum birimdeki iş yoğunluğuna rağmen doktora çalışması sürecinde bana manevi her türlü desteği veren tüm akademisyen arkadaşlarıma,

Tezin uygulama aşamasında vermiş oldukları maddi ve manevi destekten dolayı, İnönü Üniversitesi BAP birimine

Doktora eğitimim boyunca yoğunluğumu, sıkıntımı, stresimi anlayan hayatım boyunca aldığım her karar da yanımda olan kıymetli babam, annem, eşime ve çocuklarıma,

teşekkür ederim.

ONUR SÖZÜ

Doktora tezi olarak sunduđum “İmalat Sistemlerinde Derin Öğrenme Tabanlı Doku Hata Tespiti” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Hüseyin ÜZEN



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	viii
SEMBOLLER VE KISALTMALAR	x
ÖZET	xii
ABSTRACT	xiii
1. GİRİŞ	1
1.1 Literatür İncelemesi	3
1.1.1 Geleneksel hata tespit yöntemleri.....	4
1.1.2 Derin öğrenme tabanlı yüzey hata tespit yaklaşımları.....	7
1.1.2.1 Görüntü seviyesinde yüzey hata tespiti yapan yöntemler	7
1.1.2.2 Bölge seviyesinde yüzey hata tespiti yapan yöntemler	9
1.1.2.3 Piksel seviyesinde yüzey hata tespiti yapan yöntemler	11
1.2 Önerilen Sistemin Genel Çerçevesi	13
1.2.1 Tezin Amacı	14
1.2.2 Tezin Gerekçeleri	14
1.2.3 Tezin Çıktıları.....	15
1.2.4 Tezin Yenilikçi Yönü ve Ar-Ge Niteliği	16
1.2.5 Tezin Organizasyonu	16
2. DERİN ÖĞRENME.....	18
2.1 Evrişimsel Sinir Ağlarının Bileşenleri.....	19
2.1.1 Evrişim katmanı.....	19
2.1.2 Havuzlama katmanı	21
2.1.3 Yukarı örnekleme katmanı	21
2.1.4 Aktivasyon fonksiyonu.....	22
2.1.5 Tam bağlı katman	23
2.1.6 Giriş ve çıkış katmanı	23
2.1.7 Dropout ve normalizasyon katmanı.....	23
2.2 Özel Blok Yapıları.....	24
2.2.1 Kaybolan gradyan problemi	24
2.2.2 Eklemeli bağlantı: Artık (residual) bağlantı	25
2.2.3 Birleştirmeli bağlantı: Dense blok.....	26
2.2.4 Inception blok	27
2.2.5 Derinlemesine ayrılabilir evrişim	28
2.3 Dikkat Mekanizmaları	30
2.3.1 Sıkma ve uyarlama bloğu	31
2.3.1.1 Mekânsal sıkma ve kanal uyarlama.....	32
2.3.1.2 Kanal sıkma ve mekânsal uyarlama	33
2.3.1.3 Mekânsal ve kanal bazlı sıkma ve uyarlama	33
2.3.2 ViT dönüştürücüsü	34
2.3.2.1 Parçalama:	35
2.3.2.2 Projeksiyon	35
2.3.2.3 Konumsal yerleştirmeler.....	35
2.3.2.4 Öz dikkat mekanizması	36

2.3.2.5	Çok başlıklı öz-dikkat mekanizması	37
2.3.2.6	ViT dönüştürücü bloğu ve tüm bileşenleri	37
2.3.3	Swin dönüştürücü	38
2.3.4	Parça genişletme	40
2.4	Transfer Öğrenme Yaklaşımı	40
2.5	Kodlayıcı-Kod Çözücü Ağlar	42
2.5.1	Unet mimarisi	43
2.5.2	Öznitelik piramit ağı	44
3.	YÜZEY HATA TESPİTİ İÇİN KULLANILAN VERİ KÜMELERİ	46
3.1	TILDA Veri Kümesi	46
3.2	MT Veri Kümesi	47
3.3	MVTec Veri Kümesi	47
3.4	MVTec-Doku Veri Kümesi	48
3.5	DAGM Veri Kümesi	49
3.6	AITEX Veri Kümesi	49
3.7	Kullanılan Performans Ölçütleri	50
4.	YÜZEY HATA TESPİTİ İÇİN ÖNERİLEN YENİ YAKLAŞIMLARI	51
4.1	Uygulama 1: Inceptionv3 Tabanlı Zenginleştirilmiş Öznitelik Entegrasyon Ağ Mimarisi	51
4.1.1	Önerilen yöntem	51
4.1.1.1	Çok seviyeli öznitelik çıkartma modülü	52
4.1.1.2	Öznitelik piramit ağ tabanlı kod çözücü modülü	54
4.1.1.3	Öznitelik entegrasyon ve anlamlandırma modülü	55
4.1.2	Deneysel çalışmalar ve sonuçları	56
4.1.2.1	Veri kümeleri ve uygulama detayları	56
4.1.2.2	Deneysel sonuçlar	57
4.2	Uygulama 2: Derinlemesine Sıkma ve Uyarlama Bloğu Tabanlı Efficient-Unet Mimarisi	60
4.2.1	Önerilen yöntem	60
4.2.1.1	Derinlemesine sıkma ve uyarlama bloğu	61
4.2.1.2	Derinlemesine sıkma ve uyarlama blok tabanlı Efficient UNet	63
4.2.1.3	Önerilen DSUB-EUNet mimarisinin eğitim prosedürü	65
4.2.2	Deneysel çalışmalar ve sonuçları	66
4.2.2.1	Veri kümeleri ve uygulama detayları	66
4.2.2.2	MT veri kümesinin sonuçları	67
4.2.2.3	DAGM veri kümesi sonuçları	69
4.2.2.4	AITEX veri kümesi sonuçları	71
4.2.3	Tartışma ve analiz	72
4.2.3.1	Önerilen modelin önceki çalışmalarla karşılaştırılması	72
4.2.3.2	DSUB-EUNet mimarisinin bileşen analizi	74
4.3	Uygulama 3: Çok Boyutlu Öznitelik Çıkarma Tabanlı Derin Kodlayıcı-Kod Çözücü Ağ Mimarisi	76
4.3.1	Önerilen model	77
4.3.1.1	Çok boyutlu mekânsal öznitelik çıkarma	78
4.3.1.2	3B spektral ve mekânsal öznitelikler çıkarma	78
4.3.1.3	Çok girişli dikkat kapısı	80
4.3.1.4	Ters evrişim tabanlı kod çözücü	82
4.3.1.5	Model eğitimi	83
4.3.2	Deneysel çalışmalar ve sonuçları	83
4.3.2.1	Veri kümeleri ve uygulama detayları	83

4.3.2.2	MT veri kümesi	84
4.3.2.3	MVTec veri kümesi	85
4.3.3	MFE-DEDNet bileşenlerinin analizi	88
4.4	Uygulama 4: Dönüştürücü Tabanlı Melez Dikkat Kapısı ve Yeni Bir Ağ Mimarisi.....	90
4.4.1	Önerilen yöntem	91
4.4.1.1	Öznitelik çıkarma modülü	92
4.4.1.2	Melez dikkat kapısı.....	92
4.4.1.3	Öznitelik füzyon modülü	96
4.4.2	Deneysel çalışmalar ve sonuçları	100
4.4.2.1	Veri kümeleri ve uygulama detayları	100
4.4.2.2	MDK birleştirme stratejilerinin sonuçları ve analizi	101
4.4.2.3	Son teknoloji modeller ile karşılaştırma.....	104
4.5	Uygulama 5: Swin Dönüştürücü Tabanlı Çok Özellikli Entegrasyon Ağ Mimarisi.....	106
4.5.1	Önerilen model	107
4.5.1.1	Kodlayıcı	108
4.5.1.2	Kod çözücü	108
4.5.1.3	Çok özellikli entegrasyonu	109
4.5.1.4	Modelin genel görünümü ve eğitimi	110
4.5.2	Deneysel çalışmalar ve sonuçları	111
4.5.2.1	Veri kümeleri ve uygulama detayları	112
4.5.2.2	MT veri kümesi	112
4.5.2.3	MVTec-doku	114
4.5.3	Tartışma ve analiz.....	116
4.5.3.1	Önerilen ağdaki bileşenlerin analizi	116
4.5.3.2	Önerilen modelin önceki çalışmalarla karşılaştırılması.....	117
4.6	Uygulama 6: Derin Sinir Ağına Dayalı Çoklu Havuzlama ve Filtreleme Yaklaşımı.....	119
4.6.1	Önerilen model	119
4.6.1.1	Ön işleme aşaması	120
4.6.1.2	Derin öznitelik çıkarma aşaması.....	121
4.6.1.3	Sınıflandırma aşaması	121
4.6.2	Deneysel çalışmalar	122
4.6.3	Performans sonuçları	123
4.6.3.1	Transfer öğrenme ve DVM sınıflandırıcı yaklaşımlarının analizi	123
4.6.3.2	TILDA veri kümesinden alınan detaylı sınıflandırma sonuçları ..	124
4.6.3.3	Son teknoloji modeller ile karşılaştırma	126
4.6.4	Tartışma ve analiz.....	128
4.6.4.1	Önerilen modelin ön işleme adımının analizi.....	128
4.6.4.2	Önceki çalışmalar ile kıyaslanması	130
5.	SONUÇLAR VE ÖNERİLER	133
	KAYNAKLAR.....	138
	ÖZGEÇMİŞ	152

ÇİZELGELER DİZİNİ

Çizelge 2.1: Transfer öğrenme için kullanılan başlıca ağ modelleri.....	41
Çizelge 2.2: Başlıca önceden eğitilmiş ağ mimarilerinin karakteristik özellikleri [120]....	42
Çizelge 3.1: MVTec veri kümesinde bulunan sınıflar ve sınıflarda bulunan örnek sayıları [124].	48
Çizelge 4.1: InceptionV3 mimarisinin katman detayları.	53
Çizelge 4.2: ÇSÖÇ modülün elde edilen 5 farklı öznetelik haritası ve çıkış boyutları.	54
Çizelge 4.3: Inc-ZÖEA modelin performansını değerlendirmek için kullanılan veri kümeleri ve örnek sayıları.	56
Çizelge 4.4: MT, MVTec-Doku ve DAGM veri kümesi kullanılarak elde edilen performans sonuçları.	57
Çizelge 4.5: MT, DAGM ve AITEX veri kümelerinin eğitim ve test örnekleri.	67
Çizelge 4.6: MT veri kümesinden elde edilen test sonuçları.	69
Çizelge 4.7: DAGM veri kümesinden elde edilen test sonuçları.	71
Çizelge 4.8: AITEX veri kümesinden elde edilen test sonuçları.	72
Çizelge 4.9: Yüzey hata tespiti için geliştirilen yöntemler ve sonuçları.	73
Çizelge 4.10: DSUB-EUNet bileşenlerinin analizi.	74
Çizelge 4.11: MÖÇ ağının katmanları ve her katmanda kullanılan hiperparametreler.	78
Çizelge 4.12: MFE-DEDNet ağında kullanılan 3BSM modülünün hiper parametreleri.	80
Çizelge 4.13: TEKÇ ağının katmanları ve hiper parametreleri.	82
Çizelge 4.14: MT veri kümesi ile elde edilen sayısal sonuçlar.	84
Çizelge 4.15: MVTec veri kümesi ile elde edilen sayısal sonuçlar.	86
Çizelge 4.16: DEDNet, SFEDNet, Double_SFEDNet mimarilerinin parametre sayısı ve performans sonuçları.	89
Çizelge 4.17: MobileNetV2'nin katman yapısı ve parametreleri (t: genişleme değeri, n: tekrarlama sayısı, IRB: NE, DE ve NE katmanları).	92
Çizelge 4.18: Önerilen MDK-ÖFA ağının yapısı (pb: parça boyutu, C ve D: şekil 4.26'deki NE'nin filtre sayısı).	99
Çizelge 4.19: MDK-ÖFA ağlarının sürümleri ve bileşenleri.	100
Çizelge 4.20: Mevcut çalışmada kullanılan parametreler.	100
Çizelge 4.21: MDK-ÖFA modellerinin MT veri kümesindeki performans sonuçları.	102
Çizelge 4.22: MDK-ÖFA modellerinin MVTec-Doku veri kümesindeki performans sonuçları.	104
Çizelge 4.23: MT veri kümesi için MDK ^{max} -ÖFA ve son teknoloji modellerin performans sonuçları.	105
Çizelge 4.24: MVTec-Doku veri kümesi için MDK ^{max} -ÖFA ve son teknoloji modellerin performans sonuçları.	105
Çizelge 4.25: Swin-ÇÖEA ağının katman ve parametre ayarları. (PS: dönüştürücü için girdinin parça sayısı; Head: dönüştürücüdeki kullanılan dikkat başlık sayısı; mlp : MLP'de için nöron sayısı).	111
Çizelge 4.26: k=2 çapraz doğrulama için, Fold 1 ve Fold 2'de bulunan hatalı ve hatasız görüntülerin sayısı.	112
Çizelge 4.27: MT veri kümesinin test sonuçları.	114
Çizelge 4.28: MVTec-Doku veri kümesinin test sonuçları.	116
Çizelge 4.29: Swin-ÇÖEA modelindeki kod çözücü katman analizi.	116

Çizelge 4.30: Swin-MFNet modeli için farklı omurga ağları kullanılarak elde edilen sonuçlar.....	117
Çizelge 4.31: MT ve MVTec-Doku veri kümelerini kullanan önceki çalışmaların sonuçları.	118
Çizelge 4.32: Orijinal ve veri artırma ile elde edilen veri kümeleri ve detayları.....	122
Çizelge 4.33: DVM ve transfer öğrenme (ESA) tabanlı ResNet101 mimarisinin doğruluk puanları.	123
Çizelge 4.34: Önerilen DSA-ÇHF modelinin bireysel doğruluk puanları.....	125
Çizelge 4.35: Önerilen DSA-ÇHF modelinin birleştirilmiş veri kümeleri üzerindeki doğruluk puanları.....	125
Çizelge 4.36: Önerilen DSA-ÇHF modelinin diğer mimarilerle karşılaştırılması.....	127
Çizelge 4.37: Birinci deneysel çalışmanın doğruluk puanları.	129
Çizelge 4.38: Üçüncü deneysel çalışmanın doğruluk puanları.	130
Çizelge 4.39: Önerilen model sonuçlarının önceki çalışmalarla karşılaştırılması.	131



ŞEKİLLER DİZİNİ

Şekil 1.1: Yüzey hata tespitinde karşılaşılan problemlerle ilgili yüzey örnekleri ve hata konumları (kırmızı işaret); (a) yüzey ve hata çeşitliliği, (b) arka planla aşırı benzerlik ve (c) düşük kontrast.	3
Şekil 2.1: ESA mimarilerinin genel yapısı [81].	18
Şekil 2.2: Evrişim katmanında evrişim süreci.	20
Şekil 2.3: Havuzlama işlemi ile ilgili örnek gösterim.	21
Şekil 2.4: Yukarı Örnekleme Katmanı.	22
Şekil 2.5: Artık bağlantı modeli.	25
Şekil 2.6: Dense blok yapısı [91].	26
Şekil 2.7: Inception blok yapıları [96], a) Orijinal IB, b) IB1, c) IB2, d) IB3.	28
Şekil 2.8: Derinlemesine ayrılabilir evrişim (a) ve standart evrişim (b) yapıları.	29
Şekil 2.9: Sıkma ve uyarlama bloğunun temel yapısı.	31
Şekil 2.10: Mekânsal Sıkma ve Kanal Uyarlama (MSKU).	32
Şekil 2.11: Kanal Sıkma ve Mekânsal Uyarlama (KSMU).	33
Şekil 2.12: Mekânsal ve Kanal bazlı Sıkma ve Uyarlama (MKSU).	34
Şekil 2.13: Öz-dikkat mekanizmasının (a) görselleştirilmesi ve (b) hesaplanması.	37
Şekil 2.14: Dönüştürücü mekanizması.	38
Şekil 2.15: (a) Swin Dönüştürücü ve (b) pencere kaydırma modeli.	39
Şekil 2.16: Parça genişletme katmanı ayrıntıları.	40
Şekil 2.17: Kodlayıcı ve kod çözücü ağlarının genel yapısı.	43
Şekil 2.18: Unet mimarisinin genel yapısı [67]	44
Şekil 2.19: Öznitelik piramit ağ mimarisi [65], [66].	45
Şekil 3.1: TILDA veri kümesi (C1-C4 kumaş tipi ve e0-e6 hata sınıfları).	46
Şekil 3.2: MT veri kümesinden alınmış örnek görüntüler.	47
Şekil 3.3: MVTEc-Doku veri kümesi [122]	49
Şekil 3.4: DAGM veri kümesindeki bazı sınıflar ve elips şeklindeki etiketlemeleri [127].	49
Şekil 3.5: AITEX veri kümesinden alınmış örnek görüntüler.	50
Şekil 4.1: Önerilen Inc-ZÖEA mimarisi.	51
Şekil 4.2: Eklemeli atlama bağlantılarındaki öznitelik aktarma işlemi.	54
Şekil 4.3: MT veri kümelerinden alınan örnek görsel sonuçlar.	58
Şekil 4.4: MVTEc-Doku veri kümelerinden alınan örnek görsel sonuçlar.	59
Şekil 4.5: DSUB-EUNet mimarisinin temel yapısı.	61
Şekil 4.6: Önerilen DSUB yaklaşımının yapısı (<i>NE</i> :Noktasal Evrişim, <i>DE</i> : Derinlemesine Evrişim).	61
Şekil 4.7: DSUB modelinde kullanılan evrişim (a) ve Orijinal KSMU modelinde kullanılan (b) evrişim katman modelleri.	62
Şekil 4.8: MBConv Bloğunun Yapısı	64
Şekil 4.9: (a) EfficientNetB0 ağının ayrıntıları, (b) artık bağlantı.	64
Şekil 4.10: Ters Evrişim katmanı; x katmanının girdi verileri, z katmanının çıktıları ve w filtrenin ağırlık değerleridir.	65
Şekil 4.11: MT veri kümesindeki görsel sonuçları.	68
Şekil 4.12: DAGM veri kümesindeki görsel sonuçlar.	70
Şekil 4.13: AITEX veri kümesindeki görsel sonuçlar.	71
Şekil 4.14: DSUB-EUNet ve KSMU-EUNet test sonuçları.	75

Şekil 4.15: (a) 2B evrişim ve (b) 3BSM modülü.	76
Şekil 4.16: MFE-DEDNet ağının genel yapısı.....	77
Şekil 4.17: 3BSM modülünün işlem adımları.....	79
Şekil 4.18: Çok Girişli Dikkat Kapısı.	81
Şekil 4.19: MT veri kümesinden görsel sonuçlar.	85
Şekil 4.20: MVTEc veri kümesi ile elde edilen görsel sonuçlar.	88
Şekil 4.21: Önerilen MDK tabanlı Öznitelik Füzyon Ağı (MDK-ÖFA).	91
Şekil 4.22: Dönüştürücü (Trans) tabanlı dikkat Bloğu.	92
Şekil 4.23: Parça Çıkartma Bloğu.	93
Şekil 4.24: MDK modelinin dönüştürücü mekanizması (a) ve parça birleştirme bloğu.....	94
Şekil 4.25: Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu.	95
Şekil 4.26: Melez Dikkat Kapısı (MDK, Hybrid Attention Gate).	95
Şekil 4.27: Öznitelik Füzyon Modülü.	97
Şekil 4.28: MT veri kümesinden örnek görüntüler ve MDK-ÖFA modellerinin tahmin sonuçları.....	102
Şekil 4.29: MVTEc-Doku veri kümesi için önerilen ve diğer modellerin görsel tahmin sonuçları.....	103
Şekil 4.30: Önerilen Swin-ÇÖEA mimarisi.....	108
Şekil 4.31: Çok özellikli entegrasyon modülü.	109
Şekil 4.32: MT görsel sonuçları, GT: Gerçek Değer.	113
Şekil 4.33: MVTEc-Doku görsel sonuçları, GT: Gerçek Değer.....	115
Şekil 4.34: Önerilen DSA-CHF modelinin genel şeması.....	119
Şekil 4.35: Ön işleme aşamasında durum 1 için örnek gösterim.	121
Şekil 4.36: DVM ile verilerin sınıflandırılması.	122
Şekil 4.37: Önerilen modelin karışıklık matrisleri.....	126
Şekil 4.38: DSA-CHF modeli ve diğer ESA mimariler ile karşılaştırması.	128

SEMBOLLER VE KISALTMALAR

3BSM	: 3B Spektral ve Mekânsal Öznitelikler Çıkarma
B	: Boyut
BN	: Batch Normalizasyon
BTA	: Bölge Teklif Ağı
ÇÖE	: Çok Özellikli Entegrasyon
ÇSÖÇ	: Çok Seviyeli Öznitelik Çıkartma
D	: Derinlik
DAE	: Derinlemesine Ayrılabilir Evrişim
DE	: Derinlemesine Evrişim
DSUB	: Derinlemesine Sıkma ve Uyarlama Bloğu
DVM	: Destek Vektör Makinaları
ESA	: Evrişimsel Sinir Ağı
EBR	: Evrişim, Batch Normalizasyon ve ReLU
GOH	: Global Ortalama Havuzlama
GIÜ	: Grafikselle İşleme Ünitesi
H	: Yükseklik
IB	: Inception Bloğu
K	: Anahtar (Key)
KSMU	: Kanal Sıkma ve Mekânsal Uyarlama
LN	: Katman Normalizasyonu (LayerNorm)
MDK	: Melez Dikkat Kapısı
ÇGDK	: Çok Girişli Dikkat Kapısı
MKSU	: Mekânsal ve Kanal bazlı Sıkma ve Uyarlama
MLP	: Çok Katmanlı Projeksiyon (Multi Layer Perceptron)
MSA	: Çok Kafalı Öz-Dikkat (Multi-Head Self Attention)
MÖÇ	: Mekânsal Öznitelik Çıkarma
MSKU	: Mekânsal Sıkma ve Kanal Uyarlama
NE	: Noktasal Evrişim
ÖEA	: Öznitelik Entegrasyon ve Anlamlandırma
ÖPA	: Öznitelik Piramit Ağı
ÖY	: Önerilen Yöntem
P	: Dolgulama (Padding)
PÇ	: Parça Çıkartma

PG	: Parça Genişletme
Q	: Sorgu (Query)
SU	: Sıkma ve Uyarlama
SW-MSA	: Kaydırılmış Pencere Tabanlı Bir MSA (Shifted Window Based MSA)
TB	: Tam Bağlı (Fully Connected)
TBA	: Temel Bileşen Analizi
TEA	: Tam Evrişimsel Ağ (Fully Convolutional Network)
TEKÇ	: Ters Evrişim Tabanlı Kod Çözücü
TÖ	: Transfer Öğrenme
Trans	: Transformer
V	: Değer (Value)
VA	: Veri Arttırma
ViT	: Vision Transformer
W	: Genişlik
W-MSA	: Pencere Tabanlı MSA (Window Based MSA)
YÖ	: Yukarı Örnekleme
YSA	: Yapay Sinir Ağlarından
<i>pb</i>	: Parça Boyutu
<i>fb</i>	: Filtre Boyutu
<i>fs</i>	: Filtre Sayısı
<i>ad</i>	: Adım Genişliği

ÖZET

Doktora Tezi

İMALAT SİSTEMLERİNDE DERİN ÖĞRENME TABANLI DOKU HATA TESPİTİ

Hüseyin ÜZEN

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

153+XIII sayfa

2022

Danışman: Prof. Dr. Davut HANBAY

Son zamanlarda bilgisayar yazılım ve donanım teknolojisindeki gelişmelerle birlikte nesnelerin interneti, nesne sınıflandırma, nesne tespiti ve örüntü tanıma gibi kavramlar günümüzde oldukça yaygınlaşmıştır. Bu kavramlarla geliştirilen sistemler tıp, kimya, mühendislik, tarım, güvenlik sistemleri ve coğrafi bilimleri gibi birçok alanda yenilik ve kolaylık sağlamıştır. Özellikle derin öğrenme tabanlı sistemler, birçok alanda yüksek performanslar sağlamıştır. Bu tez çalışmasında, güncel derin öğrenme sistemlerini imalat sistemlerinde uyarlamak ve imalat sistemlerinde otomatik doku hata tespiti için yeni yaklaşımlarının geliştirilmesine odaklanılmıştır. Genellikle, imalat sistemlerinde yüzey/doku hata tespiti, uzman personel tarafından manuel olarak yapılmaktadır. Manuel yapılan denetimlerin zaman ve doğruluk açısından olumsuz yönleri vardır. Bunun yanı sıra uzman personel gereksiniminden dolayı maliyetlidir. Buna kıyasla, yapay zekâ destekli otomatik hata tespit sistemleri performans ve güvenilirlik açısından daha başarılı sonuçlar üretebilmektedir. Bu başarılı sonuçlar sayesinde manuel denetimlerden kaynaklanan maddi kayıplar önlenmekte ve üretim alanında güçlü kazançlar sağlanabilmektedir. Bu bağlamda bu tez çalışmasında otomatik hata tespiti için derin öğrenme tabanlı etkili ve yeni modeller önerilmiştir. Bu modeller aşağıda sıralanmıştır:

- Hata tespiti için zenginleştirilmiş öznitelik piramit ağ mimarisi,
- Önemli özellikleri güçlendirmek için Derinlemesine Sıkma ve Uyarlama Bloğu (DSUB) ve DSUB tabanlı Efficient-Unet mimarisi,
- Derin öğrenmede çok boyutlu mekânsal, spektral ve anlamsal öznitelikler çıkarmak için yeni bir yaklaşım,
- Son zamanlarda oldukça popüler olan görüntü tabanlı dönüştürücüleri yüzey hata tespitine uyarlamak için geliştirilen Melez Dikkat Kapısı,
- Swin dönüştürücü tabanlı yeni bir kodlayıcı ve kod çözücü ağ mimarisi,
- Kumaş hatalarının sınıflandırılması için Derin Sinir Ağına, Çoklu Havuzlama ve Filtrelemeye dayalı yeni bir yaklaşım

Önerilen bu özgün yöntemlerin başarısını analiz etmek için literatürde var olan ve güncel veri kümeleri kullanılmıştır. Deneysel çalışmalarda, önerilen yöntemler güncel modeller ile karşılaştırılmıştır. Yapılan kapsamlı deneysel çalışmalardaki sonuçlara göre, hata tespiti için geliştirilen yöntemlerin etkili olduğu gözlemlenmiştir.

Anahtar Kelimeler: Yüzey ve Doku Hata Tespiti, Dikkat Kapıları, Dönüştürücüler, Evrişimsel Sinir Ağları, Derin Öğrenme

ABSTRACT

Phd. Thesis

DEEP LEARNING BASED TEXTURE DEFECT DETECTION IN MANUFACTURING SYSTEMS

Hüseyin ÜZEN

Inonu University

Graduate School of Nature and Applied Sciences
Department of Computer Engineering

153+XIII page

2022

Supervisor: Prof. Dr. Davut HANBAY

Recently, concepts such as internet of things, object classification, object detection, and pattern recognition have become quite common today with the developments in computer software and hardware technology. Systems developed with these concepts have provided innovation and convenience in many fields such as medicine, chemistry, engineering, agriculture, security systems, and geographical sciences. Especially deep learning-based systems have provided high performances in many areas. This thesis focuses on adapting existing deep learning systems to manufacturing systems and developing new approaches for automatic texture defect detection in manufacturing systems. Generally, surface/texture defect detection in manufacturing systems is performed manually by expert personnel. Manual inspections have downsides in terms of time and accuracy. In addition, it is costly due to the need for expert personnel. In comparison, automatic defect detection systems supported by artificial intelligence can produce more successful results in terms of performance and reliability. Thanks to these successful results, financial losses caused by manual management can be prevented and strong profits can be achieved in the field of production. In this context, new and effective models based on deep learning have been proposed for automatic texture defect detection in this thesis. These models are listed below:

- Feature-enriched pyramid network architecture for defect detection,
- Depth-wise Squeeze and Excitation Block (DSEB) to strengthen important features and DSEB-based Efficient-Unet architecture,
- A new approach to extracting multidimensional spatial, spectral, and semantic features in deep learning,
- Hybrid Attention Gate which developed to adapt the recently very popular Vision Transformer to surface defect detection.
- A new encoder and decoder network architecture based on Swin Transformer,
- A new approach based on Deep Neural networks, Multiple Pooling, and Filtering for the classification of fabric defects

Existing and current datasets in the literature were used to analyze the success of these proposed original methods. In experimental studies, the proposed methods were compared with current models. According to the results of extensive experimental studies, it has been observed that the methods developed for defect detection are effective.

Keywords: Surface and Texture Defect Detection, Attention Gates, Transformer, Convolutional Neural Networks, Deep Learning.

1. GİRİŞ

İkinci dünya savaşından sonra seri üretimin yaygınlaşması; imalat sistemlerini, satış pazarlarını ve tüketici isteklerini hızlı bir şekilde değiştirmiştir. Ülkeler arası bağlantılarının yaygınlaşması ve ticari sınırlarının ortadan kalkması ile birlikte tüm dünyada ortak bir pazar oluşmuştur. Diğer yandan, sayısız yeni üreticiler ve tüketiciler pazara dâhil olması, üreticileri zorlu bir rekabete sokmuştur. Eski zamanlarda rekabet sadece düşük fiyat odaklı iken günümüzde, bu rekabet düşük fiyat ve kaliteli ürün endeksli olmuştur. Bu durum, imalat sistemlerinde kaliteyi oldukça önemli kılmıştır. Bu bağlamda işletmelerde sadece “ürün üretmek” değil “yüksek kalitede ürün üretmek” esas alınmıştır. Tüm bu nedenlerden dolayı imalat sistemlerinde kalite kontrol en önemli ihtiyaçlardan biri olmuştur [1, 2]

İmalat sistemlerinde uygulanan kalite kontrolünün en önemli adımlardan bir tanesi yüzey hata denetimidir. Üretim sırasında hammadde bozulması veya imalat makinelerindeki arızalar gibi çeşitli olumsuzluklar, üretilen ürünün yüzeyinde/dokusunda hataların oluşmasına neden olabilmektedir. Kumaş ve deri gibi ürünlerde yırtık, delik ve renk gibi doku hataları oluşurken çelik, mermer ve ahşap gibi ürünlerde ise çatlama, kırılma ve eğilme gibi yüzey hataları oluşabilmektedir.

İmalat sistemlerinde yüzey hatalarının tespit edilmesi, maddi kayıpların önlenmesi ve yüksek kalitede ürün sağlanması açısından oldukça önemlidir. Yüzey hataları yalnızca ürünün görünümünü etkilemekle kalmaz, aynı zamanda üretim performansı ve üretici güvenilirliği açısından da uzun vadeli olumsuz etkilere sebep olmaktadır. Bundan dolayı imalat sistemlerindeki en önemli görevlerden biri yüzey/doku hatalarının tespitidir [3–5].

İmalat sistemlerinde yüzey hatalarının tespiti, geleneksel olarak kalite kontrol alanında eğitilmiş uzmanlar tarafından yapılmaktadır. Uzmanlar, ürün yüzeyini denetleyerek herhangi bir yüzey hatası olup olmadığına karar vermektedir. Eğer yüzeyde bir deformasyon var ise hatanın yeri, boyutu ve tipi gibi mevcut hata ile ilgili istatistiksel veriler toplanmaktadır. Bu tespit yönteminin dezavantajı; uzman tarafından gözlemlenerek yapıldığı için zaman alıcı ve doğruluğunun uzman tecrübesine bağlı olmasıdır [5]. Bundan dolayı istihdam edilen her bir kalite kontrol personelinin yeterli bir eğitimden geçmesi gerekmektedir. Her ne kadar eğitimler istenen düzeyde olsa bile yeterli kalifiyedeki bir uzmanın yetişmesi için uzun zaman gerekmektedir. Ayrıca denetim performansı doğrudan personelin tecrübesine bağlıdır. Bu durum, günümüzde oldukça zorlaşan piyasa rekabetine ayak uyduran imalat sistemleri için oldukça zorlayıcı bir durum olmuştur. Öte yandan, insan

yapısı gereği oluşabilen yorgunluk ve dikkat dağınıklığı gibi nedenlerden dolayı uzmanlar tarafından yapılan denetimlerde performansı %45 kadar düşürdüğü bilinmektedir [5, 6].

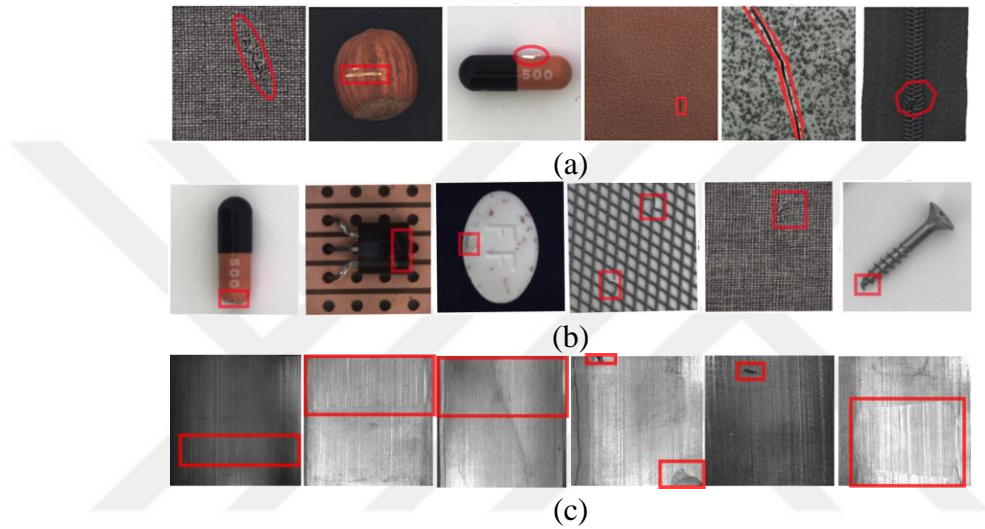
İmalat sistemlerinde yüzey/doku hatalarının tespit edilememesi ciddi maddi kayıplara yol açabilmektedir. Çünkü İmalat sistemlerindeki otomatikleşmiş üretim süreci oldukça hızlı ve tekrarlı bir mekanizmaya sahiptir. Bundan dolayı üretimin herhangi bir anında başlayan yüzey hatası sürekli olarak tekrarlanmaktadır. Bu durum çok sayıda hatalı ürün ortaya çıkmasına neden olmaktadır. Ayrıca, günümüzde sürekli devam eden üretim sistemini yavaşlatmamak için yüzey hata denetimi uzmanlar tarafından ara sıra yapılmaktadır [6–10].

Sonuç olarak imalat sistemlerinde yüzey/doku hatalarının yeterli seviyede yapılmaması veya manuel olarak yapılması aşağıdaki olumsuzluklara neden olmaktadır:

- 1) Hatalı üretim devam ettiği için ham madde, zaman ve elektrik kayıplarının oluşmasına,
- 2) Personel tarafından yapılan denetimler, uzun vadede yüksek doğruluk sağlayamadığı için üreticilerin yüksek kalitede ürün sağlayamamasına,
- 3) Hatalı ürünün piyasaya sunulmaması ya da düşük maliyetlerde sunulduğu için iş kaybının veya ekonomik zararların oluşmasına,
- 4) Her bir üretim makinesi için istihdam edilmesi gereken yeni personel bulunması ve personelin eğitilmesi gibi süreçlerden kaynaklanan zaman ve maddi kayıpların oluşmasına.

Yukarıda bahsedilen olumsuzluklar, günümüzde araştırmacıları otomatik yüzey hata tespit sistemlerini geliştirmeye yönlendirmiştir [5, 7, 11]. Otomatik yüzey hata denetimi için geliştirilen sistemlerde, kamera yardımı ile yüzey görüntüsü alınmakta ve yapay zekâ destekli algoritmalar ile kontrol edilmektedir. Hata tespiti için geliştirilen algoritmalar, görüntü içerisindeki herhangi bir kalite kaybına yol açabilecek olumsuzlukları yüzey dokusundan (arka plan) ayırmayı hedeflemektedir. Ayrıca geliştirilen algoritmaların imalat sistemlerinin yüksek hızlardaki üretim kapasitesini engellemeden gerçek zamanlı olması beklenmektedir. Bu bağlamda günümüzde birçok otomatik yüzey hata tespit çalışması yapılmıştır. Bu çalışmalarda personel tarafından yapılan manuel denetimlere göre otomatik yüzey hata tespit yöntemlerinin doğruluk, devamlılık ve güvenilirlik açısından daha etkili olduğu kanıtlanmıştır [5, 7].

Otomatik yüzey hata tespitinde karşılaşılan üç temel zorluk vardır (Şekil 1.1). Birincisi, imalat sistemlerinde yüzey ve yüzey hata türlerinin çok çeşitli olması ve oluşabilecek hatanın boyutunun belirsiz olmasıdır. İkincisi, yüzey hatalarının arka plan (hatasız yüzey) ile benzerlik göstermesidir. Üçüncüsü, imalat sistemlerinde toz, aşırı ışık vb. gibi dış faktörlerden dolayı kamera ile alınan görüntülerin düşük kontrastta olmasıdır [7, 12]. Bu problemlerin çözüme kavuşabilmesi adına otomatik yüzey hata tespiti ilgi çekici ve önemli araştırma konularından biridir. Bunun yanı sıra otomatik yüzey hata tespiti için bu problemler ile başa çıkabilecek modellerin geliştirilmesine ihtiyaç vardır.



Şekil 1.1: Yüzey hata tespitinde karşılaşılan problemlerle ilgili yüzey örnekleri ve hata konumları (kırmızı işaret); (a) yüzey ve hata çeşitliliği, (b) arka planla aşırı benzerlik ve (c) düşük kontrast.

Bu tez çalışmasında, yukarıdaki problemleri göz önüne alınarak otomatik yüzey hata tespiti için son teknoloji olan derin öğrenme yapılarına dayalı algoritmalar ve hibrit sistemler geliştirilmiştir. Buna ek olarak, var olan yöntemler kapsamlı bir şekilde incelenmiş ve yeni alternatif yaklaşımlar sunulmuştur.

Bu tez çalışmasının birinci bölümünde otomatik yüzey hata tespiti ile ilgili genel bilgiler verilmiştir. Alt başlıklarda literatürde yapılan çalışmalar detaylandırılmış ve önerilen çalışmalar, genel bir çerçeve halinde sunulmuştur.

1.1 Literatür İncelemesi

Bu bölümde, otomatik yüzey hata tespiti ile ilgili literatürde yapılan çalışmalar kapsamlı bir şekilde incelenmiştir. Son zamanlarda otomatik yüzey hata tespit yöntemleri, geleneksel

(derin öğrenme kullanmayan) yöntemler ve derin öğrenme tabanlı yöntemler olarak sınıflandırılmıştır [5, 13–15].

1.1.1 Geleneksel hata tespit yöntemleri

Geleneksel hata tespit yöntemleri, derin öğrenme kullanılmadan temel görüntü işleme ve makine öğrenmesi yöntemlerine dayalıdır. Bu yöntemler, görüntülerden elde edilen öznitelikleri karşılaştırarak veya sınıflandırarak yüzey hatalarını tespit etmektedir. Geleneksel hata tespit yöntemleri; istatistik tabanlı, spektral tabanlı, model tabanlı ve öğrenme tabanlı yöntemler olarak gruplandırılmıştır. Bu kategorilerin belirleyici özellikleri şunlardır [5, 13, 16, 17]:

- İstatistik tabanlı yaklaşımlar, yüzeyin piksel değerlerini ve istatistiksel dağılımlarını kullanmaktadır. Bu yöntemlerde, yüzey görüntülerinde farklı istatistiksel sonuçlar üreten bölgeler dikkate alınarak yüzey tespiti yapılmaktadır. Bu yöntemler genellikle eşikleme [18], eş oluşum matrisi (co-occurrence matrix) [19], yerel ikili örüntü [20] ve yönlü gradyan histogramları (histograms of oriented gradients) [21] gibi modellerin temellerine dayalıdır. Literatürde, yüzey hata tespiti için birçok istatistiksel tabanlı yaklaşım geliştirilmiş olsa da bu yaklaşımlar gürültüye karşı çok hassastır.
- Spektral tabanlı yaklaşımlar, öncelikle yüzey görüntüsünün temel doku örüntüsünü ortaya çıkartmayı hedeflemektedir. Daha sonra bu örüntü temellerini uzaysal yerleşim yaklaşımları ile genelleştirmeyi amaçlamaktadır. Bu yöntemlerin uygulandığı yüzeyler, genellikle kumaşa bulunan iplik dizilimleri veya periyodik desenleri içeren dokulardır. Fakat rastgele desenler içeren veya homojen olmayan yüzeyler için spektral yaklaşımların zayıf kaldığı tespit edilmiştir [5, 22]. Bununla birlikte geçmiş yıllarda düz veya periyodik doku içeren yüzeylerde hata tespiti için Fourier Dönüşümü [23], Gabor Filtreleri [24] ve Dalgacık Dönüşümü [25] gibi temel yaklaşımlara dayalı çok sayıda çalışma yapılmıştır [26, 27].
- Model tabanlı yaklaşımlar genellikle yüzey dokusu için temel bir model tasarlamayı hedeflemektedir. İmalat sistemlerindeki üretim hatlarında elde edilen ürünler genellikle aynı desen veya doku örüntüsünden oluşmaktadır. Bunlar tekrarlı örüntü dizisi olarak kabul edilmektedir. Bununla birlikte bu örüntü olasılıksal (stochastic) veya olasılıksal olmayan (deterministic) yaklaşımlar ile modellenabilmektedir. Elde edilen model parametrik olasılık dağılımları ile başka bir doku yüzeyi izlenip

değerlendirilmektedir. Böylece modellenen yüzey dokusu ile eşleşen bölgeler hatasız ve eşleşmeyen hatalı olarak kabul edilebilmektedir. Markov Rastgele Alan Modeli (Markov Random Field Model) [28], Aktif Kontur Modeli (Active Contour Model) [29] ve Düzgün Seyrek Ayrıştırma (Smooth-Sparse Decomposition) modeli [30] yüzey hatalarını modellemek için yaygın olarak kullanılan yöntemlerdir.

- Öğrenme tabanlı yaklaşımlarda, öncelikle yüzey görüntüsünü analiz ederek renk, doku ve istatistiksel öznitelikler elde edilmektedir. Daha sonra bu öznitelikler belli bir karar verdirici sistem yardımı ile hata denetimi yapılmaktadır. Genellikle bu yaklaşımlarda karar verici sistem olarak Destek Vektör Makinaları (DVM), karar ağaçları ve yapay sinir ağları gibi modeller kullanılmaktadır.

Geleneksel hata tespit yöntemleri ile ilgili yapılan başlıca çalışmalar şu şekildedir:

Sakhare ve dig. [31], kumaş hatalarını tespit etmek için Fourier dönüşümü, ayrık kosinüs dönüşümü, dalgacık dönüşümü ve Gabor filtrelerini kullanmışlardır. Çalışmalarında, spektral tabanlı yöntemler ve morfolojik tabanlı yöntemler karşılaştırılmıştır. Deneysel çalışmalardan elde edilen sonuçlarda, spektral tabanlı hızlı Fourier ve ayrık kosinüs dönüşümü yöntemlerinin daha yüksek performans gösterdiği belirtilmiştir.

Yapı ve dig. [32], hata tespiti için RCT-MoGG (Redundant Contourlet Transform; a finite Mixture of Generalized Gaussian) yöntemine dayalı bir yaklaşım geliştirilmiştir. Önerilen modelde, Bayesian sınıflandırıcı kullanılarak hatalı ve hatasız olarak yüzey görüntüleri sınıflandırılmıştır. Deneysel çalışmalarda TILDA veri kümesinin bazı alt sınıfları kullanılmıştır. Bu deneysel çalışmalarda %96 ve üzeri başarılar elde edilmiştir.

Liu ve Zheng [33], iki boyutlu entropi ve hiper-karmaşık Fourier dönüşümüne dayalı denetimsiz bir kumaş hata tespit yöntemi önermişlerdir. Bu modelde, öncelikle hatasız kumaş yüzeyinin düzgün dağılımı elde edilmiştir. Daha sonra kumaş hatası içeren görüntülerde bu dağılıma uygun olmayan bölgeler hatalı bölge olarak kabul edilmiştir. Deneysel çalışmalarda periyodik desenler içeren kumaş tipleri kullanılmıştır. Bu çalışmanın sonuçlarına göre, nokta deseni içeren kumaş yüzeylerinin tespiti için %95,18 doğruluk skoru elde edilmiştir.

Dong ve dig. [34], X-ışın altında alınan görüntülerdeki hataları tespit etmek için rastgele orman tabanlı bir model önermişlerdir. Bu modelde öncelikle görüntüdeki her pikselin hatalı olup olmadığına karar verilmiştir. Ardından, hatalı pikseller birleştirilerek hatanın tam

konumu tespit edilmiştir. Yapılan deneysel çalışmada, gerçek ortamdan alınan çelik kaynak görüntüleri kullanılmış ve hatalı bölgelerinin %80'i tespit edilmiştir.

Cao ve dig. [35], otomatik hata tespiti için eşikleme, bölütleme ve görüntü eşleştirme yöntemlerini birlikte uygulayan hibrit bir yapı geliştirmişlerdir. Yöntemlerinde test görüntüleri arka görüntü ve ön görüntü olmak üzere ikiye bölünmüştür. Ön görüntü, ürün logosu veya marka adı gibi özel desenlerden oluşurken, arka görüntü düz ve homojen bir yapıdadır. Arka plandaki hataları tespit etmek için özel olarak geliştirilmiş bir eşikleme yöntemi kullanılmıştır. Bu eşikleme yöntemi gradyan geçişlerine dayalıdır. Deneysel çalışmalarda sınırlı örneğe sahip veri kümeleri kullanılmış ve %96 F1-skoru elde edilmiştir.

Abid [36], otomatik hata tespiti için polinom interpolasyonu kullanarak yeni bir yaklaşım geliştirmiştir. Bu yöntemde, özel bir filtre ile her bir görüntü parçasından 4 farklı polinom katsayısı elde edilmiştir. Daha sonra bu katsayılar YSA ile değerlendirilerek hata tespiti yapılmıştır. Deneysel çalışmalarda yazar tarafından oluşturulmuş bir veri kümesi kullanılmış ve %98,6 sınıflandırma performansı elde edilmiştir.

Zhang ve dig. [37], kumaş hata sınıflandırması için Gri Seviyeli Birlikte Oluşum Matrisi (Gray-level Co-occurrence Matrix (GLCM)) ve Yerel İkili Örüntü (Local Binary Patterns (LBP)) birleştiren bir algoritma önermişlerdir. Önerilen algoritmanın en önemli noktası, hatalı kumaş görüntülerinin yerel ve global özniteliklerinin çıkarılması olarak belirtilmiştir. Önerilen algoritmada yerel öznitelik bilgisi için LBP algoritması ve global öznitelik bilgisi için GLCM algoritması kullanılmıştır. Elde edilen bu öznitelikler, kumaş hatalarını sınıflandırmak için yapay sinir ağı mimarisine aktarılmıştır. Deneysel çalışmalarda TILDA veri kümesinden seçilen 100 görüntü kullanılmış ve %97,6 doğruluk skoru elde edilmiştir.

Salem ve Abdelkrim [38], TILDA veri kümesini temel alarak kumaş hata tiplerini sınıflandırmak için DVM sınıflandırıcısını kullanmışlardır. Bu çalışmada “delik”, “bükülme”, “eksik atkı” ve “yağlı saten” gibi hataları içeren görüntüler ve hatasız görüntüler ele alınmıştır. Öznitelik çıkarma işleminde GLCM, LBP ve LPQ yöntemleri kullanılmış ve bu yöntemlerden elde edilen öznitelikler DVM yöntemi ile sınıflandırılmıştır. Deneysel çalışmalardaki sonuçlara göre en iyi doğruluk skoru GLCM ile %98,75 ve LBP yöntemi ile %97,25 olarak elde edilmiştir.

Deotale ve Sarode [39], TILDA veri kümesindeki kumaş hatalarını tespit etmek için GLCM, Gabor dalgacık dönüşümü ve rastgele karar orman yöntemlerini kullanmışlardır.

Deneysel çalışmalarda, veri kümesindeki hataları sınıflandırmak için Bire-Hepsi yaklaşımı kullanılmış ve %80,2 ile %88,2 arasında değişen doğruluk performansları elde edilmiştir.

Djukic ve Spuzic [40], hata tespiti için piksel yoğunluklarının olasılık dağılımına dayalı dinamik eşikleme yaklaşımını sunmuşlardır. Bu yaklaşımda endüstriyel çelik için sıcak haddeleme makinesinden elde edilen görüntüler kullanılmıştır. Deneysel çalışmalarda elde edilen sonuçlar, bu yaklaşımın güvenilir bir şekilde hataları tespit etme potansiyeline sahip olduğunu göstermiştir.

1.1.2 Derin öğrenme tabanlı yüzey hata tespit yaklaşımları

Geleneksel yüzey hata tespit yöntemlerinin en büyük dezavantajı yüzey hata tespiti için özniteliklerin manuel olarak ayarlanmasıdır. Başka bir deyişle, uygulanacak yüzeye bağlı olarak öznitelik çıkartma yöntemi belirlenmektedir. Bu doğrultuda, en uygun öznitelikler için çok sayıda deneysel çalışma yapılması gerekmektedir. Bunun yanı sıra, günümüzde oldukça popüler hale gelen derin öğrenmede, en uygun öznitelik çıkartma işlemi öğrenme yolu ile otomatik olarak yapılmaktadır. Bu sayede, derin öğrenme modelleri birçok alanda büyük başarılar sağlamıştır. Benzer şekilde, yüzey hata tespiti için derin öğrenme tabanlı yöntemler kullanılarak üstün performanslar elde edilmiştir [7, 11].

Derin öğrenme tabanlı yüzey hata tespit yöntemleri; görüntü seviyesinde, bölge seviyesinde ve piksel seviyesinde olmak üzere üç hata tespiti olarak ayrılmaktadır [7, 41, 42].

1.1.2.1 Görüntü seviyesinde yüzey hata tespiti yapan yöntemler

Görüntü seviyesinde hata tespit yöntemlerinde, genellikle AlexNet ve VGGNet gibi sınıflandırıcı tabanlı Evrişimsel Sinir Ağ (ESA) mimarileri kullanılarak görüntüde hata olup olmadığına karar verilmektedir. Bununla birlikte bazı yöntemlerde otomatik öznitelik çıkartmak için evrişim katmanları kullanılırken sınıflandırıcı için DVM gibi geleneksel makine öğrenme yöntemleri kullanılmıştır [7, 14].

Görüntü seviyesinde hata tespiti ile ilgili yapılan başlıca çalışmalar şu şekildedir.

Masci ve dig. [43], yüzey hatalarını sınıflandırmak için 5 katmandan oluşan bir ESA mimarisi tasarlamışlardır. Önerilen ESA mimarisini optimize etmek için yedi farklı hata tipi içeren bir veri kümesi kullanılarak deneysel çalışmalar gerçekleştirilmiştir. Bu sonuçlara göre, önerilen ESA mimarisi farklı öznitelikler ile desteklenmiş DVM tabanlı yaklaşımlardan daha iyi performans elde ettiği gözlemlenmiştir.

Weimer ve dig. [44], yüzey hata tespiti için 5 ile 11 arasında değişen farklı katman sayılarına sahip birkaç ESA mimarisi geliştirilmiştir. Bu mimariler, 6 farklı desene sahip sentetik yüzey görüntüsü ve hata tipi kullanılarak analiz edilmiştir. Elde edilen sonuçlarda, ESA tabanlı yöntemlerin geleneksel yöntemlerden daha iyi performans gösterdiği tespit edilmiştir.

Jing ve dig. [45], kumaş hata tespiti için ESA tabanlı bir model önermişlerdir. Önerilen modelde, öncelikle kumaş görüntüleri parçalara ayrılmıştır. Daha sonra her kumaş parçası etiketlenmiştir. Bunun yanı sıra parça boyutunu belirlemek için mesafe eşleştirme fonksiyonu (distance matching function) uygulanmıştır. Bu şekilde benzer ve tekrarlayan parçalar elde edilmiştir. Elde edilen parçalar, önceden eğitilmiş bir ESA mimarisi kullanılarak sınıflandırılmıştır. Kullanılan ESA mimarisinin başlangıç parametreleri için LeNet-5, AlexNet ve VGG16 gibi önceden eğitilmiş ağ mimarileri kullanılmıştır. Deneysel çalışmalarda ortalama %97,31 doğruluk skoru elde edilmiştir.

Li ve dig. [46], hata tespiti için Inception blok yapılarına dayalı mikro mimariler önermişlerdir. Önerilen mikro mimariler, temel olarak farklı çekirdek boyutlarına sahip evrişim katmanları içermektedir. Ayrıca, yapılan çalışmada hata tespit uygulamaları için bu mikro mimarilerin farklı kombinasyonu ile WACNet mimarisi oluşturulmuştur. Deneysel çalışmalarda, yazarlar tarafından oluşturulmuş bir veri kümesi kullanılmıştır. Elde edilen sonuçlara göre düşük parametre sayısına sahip WACNet mimarisi ile AlexNet, VGG-16 ve GoogLeNet mimarilerine göre daha yüksek doğruluk (%97,82) elde edilmiştir.

Mei ve dig. [47], kumaş hatalarının tespiti için otomatik kodlayıcı tabanlı MS-CDAE (Multi-scale Convolutional Denoising Autoencoder) ağı önermişlerdir. Bu çalışmada, öncelikle bu ağ mimarisi kullanılarak kumaş görüntüleri yeniden oluşturulmuştur. Daha sonra oluşturulan görüntüler, orijinal görüntüler ile karşılaştırılmıştır. Bu karşılaştırma sonuçlarına göre, bir eşikleme işlemi uygulanmış ve yüzey hataları tespit edilmiştir. Deneysel çalışmalarda Fabrics, KTH-TIPS, Kylberg Texture ve ms-Texture veri kümeleri kullanılmıştır. Sonuç olarak, %80 ile %84 arasında doğruluk skorları elde edilmiştir.

Yi ve dig. [48], çelik yüzey hatalarını sınıflandırmak için, 14 katmanlı bir ESA mimarisi geliştirmişlerdir. Önerilen mimaride, evrişim ve havuzlama katmanlarından sonra tam bağlı katmanlar uygulanmıştır. Ağ mimarisinin çıktısı, 7 farklı görüntü hatasını kategorize edecek şekilde düzenlenmiştir. Deneysel çalışmalarda Northeastern University (NEU) yüzey veri kümesi kullanılmış ve DVM sınıflandırıcı kullanan bir model ile kıyaslanmıştır. Elde edilen

sonuçlara göre önerilen model %99,05 doğruluk skoru elde edilirken DVM modelinin doğruluk skoru %98,93'tür.

Cheon ve dig. [49], K en yakın komşu (K-Nearest Neighbors (KNN)) ve ESA yöntemlerinin kombinasyonuna dayalı bir sistem geliştirmişlerdir. Bu sistemde, taramalı elektron mikroskop görüntüleri kullanılarak gofret yüzey (wafer surface) hataları sınıflandırılmıştır. Deneysel çalışmalarda yazarlar tarafından oluşturulmuş bir veri kümesi kullanılmış ve %96,2'lik bir sınıflandırma doğruluğu elde edilmiştir. Öte yandan yapılan çalışmada, yeni hata tiplerinin belirlenmesi için KNN modelinin etkili olduğu vurgulanmıştır.

Masci ve dig. [50], çelik üzerindeki hataları sınıflandırmak için Çok Ölçekli Piramidal Havuzlama Ağ mimarisi önermişlerdir. Önerilen ağ mimarisi farklı boyutlardaki görüntüleri işleyebilen piramidal havuzlama katmanı içermektedir. Piramidal havuzlama katmanında, farklı boyuttaki görüntüler havuzlama ve evrişim katmanları ile sabit boyuta getirilmiştir. Sabit boyuta getirilmiş görüntüler ESA mimarisi ile sınıflandırılmıştır. Deneysel çalışmalarda önerilen model etkili sonuçlar üretmiş ve önceki çalışmalarda kullanılan modellere göre daha yüksek skorlar elde etmiştir.

1.1.2.2 Bölge seviyesinde yüzey hata tespiti yapan yöntemler

Bölge seviyesinde hata tespiti yapan yöntemler, görüntülerdeki hataları bir dikdörtgen ile temsili olarak tespit etmektedirler. Bu dikdörtgen; hatalı bölgeyi kapsayacak şekilde tanımlanmaktadır. Bölge seviyesinde hata tespiti için genellikle R-ESA [51], Faster R-ESA [52], SSD [53] ve YOLO [54] gibi yöntemler kullanılmıştır [55, 56].

Bölge seviyesinde hata tespiti ile ilgili yapılan başlıca çalışmalar şu şekildedir.

He ve dig. [57], çelik hata tespiti için Bölge Teklif Ağı (BTA, Region Proposal Network) kullanarak uçtan uca bir ESA mimarisi geliştirilmiştir. Önerilen modelde öncelikle önceden eğitilmiş ResNet34/50 ESA mimarisi ile başlangıç öznetelikleri elde edilmiştir. Daha sonra ESA'nın farklı katmanlarından gelen öznetelikleri birleştirmek için bir öznetelik füzyon ağı kullanılmıştır. Son olarak, birleştirilen öznetelik haritalarında aday bölgeler çıkarmak için BTA kullanılmış ve aday bölgeler sınıflandırılmıştır. Önerilen modeli test etmek için özgün olarak oluşturulan NEU-DET veri kümesi kullanılmıştır. Deneysel çalışmalarda elde edilen sonuçlara göre, ResNet34 ve Resnet50 modelini kullanan mimariler sırasıyla %74,8 ve %82,3 ortalama AP skorları elde edilmiştir.

Yanan ve dig. [58], ray yüzey hata tespiti için YOLOv3 [59] modelini kullanmışlardır. Bu çalışmada, öncelikle yüzey görüntüsü parçalara ayrılmıştır. Daha sonra her bir parçaya ayrı ayrı YOLOv3 modeli uygulanmıştır. Önerilen modelde sınırlayıcı kutunun nesne puanını tahmin etmek için lojistik regresyon kullanılmıştır. Bunun yanı sıra sınırlayıcı kutunun sınıfını belirlemek için ikili çapraz entropi kayıp fonksiyonu uygulanmıştır. Deneysel çalışmalarda yazarlar tarafından oluşturulan ve tren ray görüntüleri içeren bir veri kümesi kullanılmıştır. Sonuç olarak, %97 üzerinde hata tespit başarısı elde edilmiştir.

Yuan ve dig. [60], MobileNetv2 ve YOLOv3 ağ mimarilerinin kombinasyonuna dayalı MOLO olarak adlandırılan bir ağ mimarisi geliştirilmiştir. Önerilen modelde, MobileNetv2 ağ mimarisinden alınan öznelik haritaları YOLOv3 ağ yapısı ile birleştirilmiştir. Deneysel çalışmada, önerilen MOLO mimarisinin YOLO mimarisine göre daha yüksek skorlara ulaşıldığı tespit edilmiştir.

Li ve dig. [61], çelik şeritlerdeki yüzey hatalarını gerçek zamanlı olarak tespit etmek için YOLO ağ mimarisinden esinlenerek yeni bir ağ mimari önerilmiştir. Bu mimari, derin öznelik elde etmek için ilk 25 katmanı ve sınırlayıcı kutu tahmini için son 2 evrişim katmanı olmak üzere toplam 27 evrişim katmanından oluşmaktadır. Deneysel çalışmalarda yazarlar tarafından geliştirilen ve 6 farklı çelik hatasını içeren bir veri kümesi kullanılmıştır. Önerilen yaklaşım kullanılarak ortalama %95,86 duyarlılık (Recall) skoru üretmiştir. Bu sonuçlara göre, önerilen gelişmiş YOLO mimarisinin, orijinal YOLO mimarisine göre daha yüksek performans sağladığı gözlenmiştir.

Li ve dig. [62], gerçek zamanlı hata tespiti için SSD ile MobileNet mimarisinin kombinasyonuna dayalı bir mimari önermişlerdir. Önerilen yaklaşımda SSD mimarisi hata tespiti için optimize edilmiştir. Geliştirilen model, dolum hattındaki cam şişelerde oluşan gedikler, oyuklar, çapaklar ve aşınmalar gibi yüzey hatalarının tespiti için uygulanmıştır. Deneysel çalışmalarda önerilen model ele alınan bazı ESA mimarileri ve geleneksel yüzey hata tespit modelleri ile kıyaslanmış ve önerilen modelin daha başarılı olduğu tespit edilmiştir.

Young ve dig. [63], çelik korozyonu gibi yüzey hatalarını tespit etmek için Daha Hızlı R-ESA (Faster R-CNN) mimarisi kullanılarak kapsamlı deneysel çalışmalar yapmışlardır. Bu çalışmalarda, yazarlar tarafından oluşturulmuş bir veri kümesi kullanılmıştır. Sonuç olarak, optimize edilmiş daha hızlı R-ESA mimarisi diğer modellere göre daha yüksek başarılar ulaşmıştır.

1.1.2.3 Piksel seviyesinde yüzey hata tespiti yapan yöntemler

Piksel seviyesinde hata tespitine dayalı geliştirilen yöntemler, görüntülerdeki hataların tam olarak konumlarını piksel seviyesinde tespit etmektedir. Bu yaklaşımda, bölge seviyesinde hata tespit yaklaşımına göre hata konumu net olarak ortaya koyulmaktadır. Bu sayede küçük veya düzensiz yüzey hataları net bir şekilde tespit edilebilmektedir. Piksel seviyesinde yüzey hata tespiti için genellikle LinkNet [64], Öznitelik Piramit Ağı (ÖPA, Feature Pyramid Networks) [65, 66] ve U-net [67] gibi Kodlayıcı-Kod çözücü (Encoder-Decoder) ağ mimarileri kullanılmıştır [7, 55, 68].

Piksel seviyesinde hata tespiti ile ilgili yapılan başlıca çalışmalar şu şekildedir:

Huang ve dig. [69], manyetik karo yüzey hatalarının tespiti için MCuePush U-Net olarak adlandırılan yeni bir mimari geliştirmişlerdir. Önerilen mimari MCue, U-Net ve Push ağ bileşenlerinden oluşmaktadır. MCue, giriş görüntüsüne geleneksel görüntü işleme modelleri ile hata konumunu kabaca belirlemek için kullanılmıştır. Daha sonra, piksel seviyesinde hata konumu için U-Net mimarisi uygulanmıştır. Bunun yanı sıra U-net çıktısını desteklemek için Push ağı kullanılmıştır. Push ağı tam bağlı katmandan oluşturulmuştur. Önerilen modeli test etmek için yazarlar tarafından oluşturulan MT veri kümesi kullanılmıştır. Deneysel çalışmalardan elde edilen sonuçlara göre, MCuePush U-Net mimarisi (%98,5 AUC skoru) U-net ve diğer geleneksel modellerden daha yüksek performans elde ettiği gözlemlenmiştir.

Aslam ve dig. [70], titanyum kaplı metal yüzeylerde oluşan hataları tespit etmek için görüntü işleme teknikleri ile desteklenmiş U-net tabanlı bir sistem önermişlerdir. Önerilen sistemde öncelikle bir takım ön işleme adımları uygulanmıştır. Ön işleme aşamasında ilk olarak yüksek çözünürlüğe sahip girdi görüntüsü 128x128x3 olarak parçalanmıştır. Daha sonra her parçaya medyan filtresi uygulanmıştır. Elde edilen parçalar yüzey hata tespiti için U-net mimarisine aktarılmıştır. U-net mimarisinin çıktısı 128x128 boyutundaki hata tespit tahmin haritasıdır. Önerilen sistemin son aşamasında hata tespit tahmin haritalarına, gürültü gidermek için morfolojik işlemler uygulanmıştır. Daha sonra nihai hata tespit tahmin haritası için parçalı tahmin haritaları birleştirilmiştir. Deneysel çalışmalarda önerilen model %91,67 F1-skora ulaşmıştır.

Dong ve dig. [7], piksel seviyesinde yüzey hata tespiti için küresel bağlam dikkat ve piramit öznitelik füzyon modüllerinden oluşan PGA-Net mimarisi önermişlerdir. Bu mimaride öncelikle başlangıç öznitelikleri çıkarmak için önceden eğitilmiş VGG16 ağı kullanılmıştır. Daha sonra VGG16'dan elde edilen öznitelikleri birleştirmek için piramit

öznitelik füzyon modülü kullanılmıştır. Bu modülde farklı seviyelerden alınan düşük ve yüksek seviyeli öznitelik haritalarına küresel bağlam dikkat kapıları uygulayarak özniteliklerin etkili bir şekilde birleştirilmesi sağlanılmıştır. Deneysel çalışmalarda NEU-Seg, DAGM, MT ve Road_defect veri kümeleri kullanılmıştır. Elde edilen sonuçlarda, önerilen model ile NEU-Seg: %82,15, DAGM 2007: %74,78, MT_defect: %71,31, Road_defect: %79,54 mIoU skorlarına ulaşılmıştır.

Cao ve dig. [41], hata tespiti için Deep Feature Fusion (DFF) ağ mimarisini önermişlerdir. Bu modelde Resnet50 ağ mimarisi kullanılarak farklı seviyelerde başlangıç öznitelik haritaları elde edilmiştir. Daha sonra farklı seviyelerden elde edilen bu öznitelik haritalarını birleştirmek için dikkat kapılarını içeren birleştirme modülü kullanılmıştır. Önerilen ağın çıktısında, çok sayıda tahmin sonucu üretilmiştir. Bu tahmin haritalarından nihai tahmin sonucu üretmek için, çok seviyeli denetim (multilevel supervision) modülü kullanılmıştır. Deneysel çalışmalarda MT, RSDD ve CFD veri kümeleri kullanılmış ve önerilen model ile MT: %73,7, RSDD: %85, CFD: %60,1 mIoU skorlarına ulaşılmıştır.

Qiu ve dig. [55], yüzey hata tespiti için High-Efficiency Fully Convolutional Networks (HE-FCN, Yüksek Verimli Tam Evrişimsel Ağlar) olarak adlandırılan yeni bir mimari sunmuşlardır. Önerilen mimari üç aşamadan oluşmaktadır. İlk adımda FCN modülü ile derin öznitelikler elde edilmiştir. İkinci adımda, elde edilen öznitelikler sınıflandırıcı ve konum tahmini için iki farklı bölüme aktarılmıştır. Son adımda ise elde edilen öznitelik haritaları evrişim katmanı uygulanarak sınıflandırıcı ve konum tahmin sonuçları elde edilmiştir. Ayrıca, önerilen modelde düşük parametreye sahip bir ağ tasarlamak için derinlemesine ayrılabilir evrişim katmanları kullanılmıştır. Deneysel çalışmalarda DAGM veri kümesi kullanılarak %73,26 mIoU skoru elde edilmiştir.

Cen ve dig. [71], yüzey hata tespit için RWSLDC (Robust Weakly Supervised Learning Of Deep CNN Framework) olarak adlandırılan yeni bir sistem önermişlerdir. RWSLDC modelinde hata sınıflandırması için karar ağaçları kullanılırken, hata tespiti için Otsu eşikleme yöntemi kullanılmıştır. Önerilen sistemde öncelikle bir ESA mimarisi ile derin öznitelikler elde edilmiştir. Daha sonra SA-CAM olarak adlandırılan bir dikkat kapısı ile öznitelikler yeniden şekillendirilmiştir. Geliştirilen SA-CAM dikkat kapısı temel olarak sıkma ve uyarlama blok yapısı ile benzer bir yapıdadır. Bu dikkat kapısında önemli ayrıntılar güçlendirilmiştir. Son olarak, SA-CAM çıktısı, hata tespit için otsu eşikleme ve sınıflandırma için karar ağaçları olmak üzere iki farklı biçimde kullanılmıştır. Deneysel

sonuçlarda, önerilen model birçok farklı veri kümesinde üstün performans sağladığı gözlemlenmiştir.

Defard ve diğ. [72], denetimsiz hata tespiti için PaDiM olarak adlandırılan bir model geliştirmişlerdir. Bu modelde, hatasız görüntüye dayalı eğitilen ESA mimarisi kullanılarak görüntülerden derin öznitelik haritaları çıkarılmıştır. Daha sonra elde edilen öznitelik haritasından gömme (embedding) vektörleri elde edilmiştir. Son olarak, hatalı ve hatasız görüntüleri ayıklamak ve bir anormallik (anomaly) skoru atamak için Mahalanobis mesafesine dayalı gömme vektörleri karşılaştırılmıştır. Deneysel sonuçlarda, MVTEC veri kümesi kullanılmış ve önerilen model ortalama %92,2 AUC skoruna ulaşmıştır.

Rudolph ve diğ. [73], hata tespiti için bir DifferNet mimarisi önermişlerdir. Bu modelde, öncelikle bir ESA modeli kullanılarak derin öznitelikler çıkartılmıştır. Daha sonra, sırasıyla bu özniteliklere normalleştirme akış algoritmasını ve eşikleme uygulanarak hatalı bölgeler tespit edilmiştir. Geliştirilen model MT ve MVTEC veri kümeleri kullanılarak test edilmiştir. Deneysel çalışma sonuçlarına göre önerilen model kullanılarak MT’de %97,7 ve MVTEC’de %94,9 ortalama AUC skorlarına ulaşılmıştır.

Wang ve diğ. [74], sınırlı öznitelik, dinamik arka plan ve gürültü içeren görüntülerde yüzey hata tespiti için Dışbükey Olmayan Toplam Varyasyon Düzenliliğine (Non-convex Total Variation Regularized) dayalı KRPCA-NTV mimarisi önermişlerdir. Bu mimaride, “Kernel Canonical Correlation Analysis” olarak adlandırılan model kullanılarak öznitelikler çıkartılmıştır. Daha sonra “Nonconvex Total Variation” ile görüntüde bulunan hatalı bölgeler tespit edilmiştir. Son olarak, belirginlik atama modeli ile hatalı bölgeler belirginleştirilmiştir. Deneysel çalışmalarda önerilen model kullanılarak MVTEC’de %54,29 ve MT’de %61,75 F1 skorları elde edilmiştir.

1.2 Önerilen Sistemin Genel Çerçevesi

Günümüzde oldukça hızlı üretim yapan imalat sistemlerinde kalite kontrolü oldukça önemli bir yere sahiptir. Yetersiz yapılan kalite kontrol denetimleri, hatalı ürünlerin üretilmesine sebep olmaktadır. Üretilen hatalı ürünler, imalat sistemleri için maddi kayıplar ve zaman kaybı oluşturacağı gibi üretici güvenilirliği açısından da olumsuz etkilere neden olmaktadır. Bunun yanı sıra, genellikle insan gözetiminde yapılan manuel hata denetimleri, zor, zaman alıcı ve karmaşık süreçler içerdiği bilinmektedir. Bu amaç doğrultusunda, imalat sistemlerinde, görüntü işleme ve makine öğrenmesi ile geliştirilen otomatik yüzey hata tespit

sistemlerine ihtiyaç bulunmaktadır. Bu tez çalışmasında otomatik yüzey hata tespiti için özgün derin öğrenme tabanlı modeller geliştirilmiştir.

Bu bölümde, bu tez çalışmasında ortaya konulan amaçlar, gerekçeler, hedefler ve beklenen çıktılar alt başlıklar halinde verilmiştir. Ayrıca otomatik yüzey hata tespitine yönelik geliştirilen sistemin amaçları, Ar-Ge niteliği, teknolojik yönü ve tez çalışmasının organizasyonu hakkında bilgiler sunulmuştur.

1.2.1 Tezin Amacı

Bu tez çalışmasının amacı, yüzey hata tespiti için derin öğrenme tabanlı yeni yöntemlerin ve algoritmaların geliştirilmesidir. Bununla birlikte yeni geliştirilen özgün yöntemlerin literatüre kazandırılması hedeflenmektedir. Bu doğrultuda yüzey hata tespiti için mevcut yöntemler detaylı olarak incelenmiş ve hata tespit performansını arttırmak için derinlemesine ayrılabilir evrişim, sıkma-uyarlama blokları, önceden eğitilmiş ESA mimarileri ve dönüştürücü tabanlı yeni ve özgün derin öğrenme modelleri geliştirilmiştir. Yüzey hatalarının neden olduğu verim kaybının önlenmesinde için geliştirilen bu özgün derin öğrenme tabanlı otomatik hata tespit mimarileri, uzman kişilere yardımcı olacak ve hata tespitinin daha kolay ve daha kısa sürede gerçekleştirilmesine imkân sağlayacaktır.

1.2.2 Tezin Gerekçeleri

Bu tez kapsamında, otomatik yüzey hata tespiti için hızlı, etkili ve yüksek performans sağlayan özgün derin öğrenme tabanlı yöntemler geliştirilmiştir. Tezin gerekçeleri şu şekildedir:

- Günümüzde hızlı üretim yapan ve yüksek kalitede ürünler sağlayan imalat sistemleri için hata denetimi oldukça önemlidir. Giriş bölümünde detaylandırıldığı gibi imalat sistemlerinde personel tarafından yapılan manuel hata denetimleri zaman alıcı ve güvenilirliği sınırlı olduğundan otomatik yüzey hata tespiti yapabilecek sistemlere ihtiyaç vardır. Bu durum tez konusunu seçmede etkili olmuştur.
- Otomatik yüzey hata tespiti ile ilgili literatürde çalışmalar hızla artmaktadır. Bu durum, bu konunun popüler olduğunu göstermektedir. Konunun popülerliği tez konusunu seçmede etkili olmuştur.
- Her geçen gün farklı derin öğrenme tabanlı yöntemler geliştirilmektedir. Bu bağlamda var olan otomatik yüzey hata tespit sistemlerinin de geliştirilmesine ihtiyaç vardır. Tez kapsamında, yüzey hata tespit ve sınıflandırma performansını arttıracak

özgün ve güncel derin öğrenme tabanlı yöntemler geliştirme isteği tez konusu seçiminde etkili olmuştur.

- Son olarak daha kısa sürede doğru hata tespitinin yapılması, otomatik yüzey hata tespit performansı açısından önemli olmasından dolayı yine tez konusu seçiminde etkili olmuştur.

1.2.3 Tezin Çıktıları

Bu tez çalışmasında, otomatik yüzey hata tespiti için derin öğrenmeye dayalı özgün ve güncel yöntemler geliştirilmiştir. Geliştirilen modeller ile otomatik yüzey hata tespiti için daha hızlı ve daha yüksek performansta hata sınıflandırma ve tespiti yapılmıştır. Tez çalışması sonucunda, tezin çıktıları akademik olarak ele alınmıştır. Önerilen derin öğrenme yöntemleri ve gerçekleştirilen uygulamalar sonucunda hata tespiti için yapılan bilimsel çalışmalar aşağıda listelenmiştir.

1. Tez kapsamında yapılan ve SCI kapsamındaki dergilerde yayınlanan makaleler:

- Uzen, H., Turkoglu, M., & Hanbay, D. (2021). Texture defect classification with multiple pooling and filter ensemble based on deep neural network. *Expert Systems with Applications*, 175, 114838.
- Üzen, H., Türkoğlu, M., Yanikoglu, B., & Hanbay, D. (2022). Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects. *Expert Systems with Applications*, 209, 118269.
- Üzen, H., Turkoglu, M., Aslan, M., & Hanbay, D. (2022). Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection. *The Visual Computer*, 1-20.
- Üzen, H., Turkoglu, M., & Hanbay, D. (2022) Multi-dimensional Feature Extraction based Deep Encoder-Decoder Network for Automatic Surface Defect Detection, *Neural Computing and Applications*, 1-20
- Üzen, H., Turkoglu, M., Arı, A., & Hanbay, D. (2023). Piksel seviyesinde yüzey hata tespiti için InceptionV3 tabanlı zenginleştirilmiş öznetelik entegrasyon ağ mimarisi, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 2023, 38(2), 721-732

- Üzen, H., Turkoglu, M., Öztürk, D., & Hanbay, D. H. “A Novel Hybrid Attention Gate Based on Vision Transformer for the Detection of Surface Defects”, Signal, Image and Video Processing, Hakem değerlendirmesinde

2. Tez kapsamında yapılan ve alan indeksi kapsamındaki dergilerde yayınlanan makaleler:

- Üzen, H., Turkoglu, M., & Hanbay, D. (2021). Result Weighting-Based Resnet Feature Pyramid Network Architecture for Surface Defect Detection. Gazi University Journal of Science Part C: Design and Technology, 9(4), 760-772.
- Üzen, H., Turkoglu, M., Sel, İ., & Hanbay, D. (2021). Derinlemesine Özellik Piramit Ağı Kullanarak Yüzey Hata Tespiti. Computer Science, (Special), 109-115.

3. Tez kapsamında yapılan ve Uluslararası sempozyumlarda yayınlanan bildiriler:

- Üzen, H., Turkoglu, M., Karci, A., & Hanbay, D. (2019, Sep). Automatic Thresholding Method Developed With Entropy For Fabric Defect Detection, in 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), pp. 1–4.
- Üzen, H., Turkoglu, M., & Hanbay, D. (2021, jun). Surface defect detection using deep U-net network architectures”, SIU 2021- 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings, Jun. 2021.

1.2.4 Tezin Yenilikçi Yönü ve Ar-Ge Niteliği

Tez çalışması kapsamında geliştirilen derin öğrenme tabanlı yöntemler, literatürde sıklıkla kullanılan veri kümeleri ile test edilmiştir. Önerilen bu yöntemler, son teknoloji yöntemlere kıyasla daha başarılı yüzey hata tespit ve sınıflandırma sonuçlarının elde edildiği görülmektedir. Ayrıca elde edilen sonuçlarda, geliştirilen yöntemlerin etkili, güvenilir ve zaman açısından daha iyi ve yüksek doğrulukta çalıştığı tespit edilmiştir. Sonuç olarak, otomatik yüzey hata tespiti için geliştirilen yeni derin öğrenme tabanlı yöntemler tezin en büyük yeniliği olarak görülebilmektedir.

1.2.5 Tezin Organizasyonu

Bu tez çalışması beş bölümden oluşmaktadır. Bölüm 1’de, tez konusu hakkında bilgiler verilmiş olup, kapsamlı literatür araştırması yapılmıştır. Tezin amacı, gerekçeleri, çıktıları ve literatüre yapılmış olan katkıları açıkça belirtilmiştir. Son olarak tezin organizasyonu yine bu bölümde sunulmuştur.

Bölüm 2’de, tez kapsamında kullanılan derin öğrenme ve teorik altyapısı detaylı bir şekilde ele alınmıştır.

Bölüm 3’te tez kapsamında otomatik yüzey hata tespiti için kullanılan veri kümeleri detaylı bir şekilde açıklanmıştır.

Bölüm 4’te otomatik yüzey hata tespiti için önerilen derin öğrenme tabanlı yöntemler ve bu yöntemlerle gerçekleştirilen uygulamalar sonucunda elde edilen sonuçlar sunulmuştur.

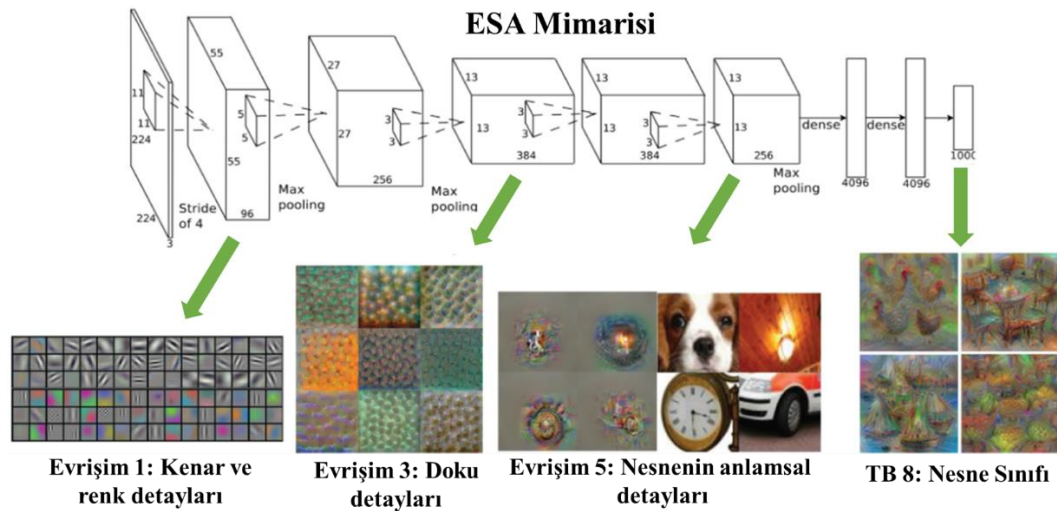
Bölüm 5’te, tez çalışması kapsamında önerilen yöntemler ile gerçekleştirilen çalışmalardan elde edilen sonuçlar tartışılmış olup, literatüre özgün katkıları belirtilmiştir. Ayrıca, gelecekte yapılması planlanan çalışmalar hakkında da bilgilere yer verilmiştir.



2. DERİN ÖĞRENME

Derin öğrenme beynin yapısal ve işlevsel modelinden esinlenilerek tasarlanmış bir makine öğrenmesi algoritmasıdır. Derin öğrenmenin geleneksel Yapay Sinir Ağlarından (YSA) temel farkı katmanlar arasında bağlantı biçimi ve eğitilmesi gereken parametre sayısının milyon mertebesinde olmasıdır. Bunun yanı sıra bu parametrelerin başarılı bir biçimde eğitilmesi için büyük miktarda verilere ve hesaplama gücü yüksek donanım birimlerine ihtiyaç vardır. Günümüzde gelişen teknoloji sayesinde verilerin dijitalleşmesi ve bu verilerin toplanması oldukça kolaylaşmıştır. Bunun yanı sıra birçok alanda kullanılan hesaplama gücü yüksek Grafikselle İşleme Ünitesi (GIÜ) sayesinde derin öğrenme parametrelerinin eğitilmesi mümkün olmuştur [75, 76].

Derin öğrenmenin içerdiği evrişim katmanları, havuzlama katmanları ve doğrusal olmayan bağlantılar sayesinde birçok vizyon görevinde en yüksek başarılar elde edilmiştir. Derin öğrenmenin en temel modeli olan Evrişimsel Sinir Ağ (ESA) mimarilerinin genel yapısı Şekil 2.1’de gösterildiği gibi, yinelemeli ve sıralı evrişim, ReLU, havuzlama katmanlarından oluşmaktadır. Bu katmanlar sayesinde geleneksel makine öğrenme yöntemlerine göre uygun ve etkin öznitelikler (evrişim katmanlarının eğitimi sırasında) otomatik olarak elde edilmektedir [77–80].



Şekil 2.1: ESA mimarilerinin genel yapısı [81].

ESA mimarilerinin (Şekil 2.1) ilk katmanlarında kenar, renk, doku gibi mekânsal bilgileri taşıyan düşük seviyeli öznitelikler elde edilirken, son katmanlarda güçlü ve anlamsal öznitelikler elde edilmektedir [7]. Birçok sınıflandırma probleminde ESA'nın son

katmanlarından elde edilen anlamsal öznitelikler kullanılmaktadır [82]. Bunun yanı sıra bölütleme veya piksel sınıflandırma problemleri için geliştirilen Unet ve FPNet gibi ağ mimarileri çok seviyeli öznitelik haritaları kullanılmıştır [7, 66, 67, 83]. Bu ağ mimarilerinde ayrıca atlama bağlantıları içermektedir. Bu bağlantılar ile ilk katmanlarda bulunan kenar, renk ve doku gibi mekânsal detayları içeren düşük seviyeli öznitelikler, son katmanlardaki anlamsal detayları (semantic details) içeren yüksek seviyeli öznitelikler ile birleştirmiştir. Bu bakış açısıyla geliştirilen kodlayıcı-kod çözücü ağlarda düşük seviyeli öznitelikler daha fazla kullanılarak birçok segmentasyon probleminde üstün başarı sağlamıştır [84–87].

ESA mimarilerinin temel yapı taşları Evrişim, Havuzlama, Aktivasyon ve tam bağlı katmanlardır. Bunun yanı sıra bazı gelişmiş ESA mimarilerinde artık blok, Inception ve sıkma-uyarlama gibi özel blok yapıları geliştirilmiştir. Bu bileşenleri kullanarak birçok farklı ESA mimarisi önerilmiştir [76, 88].

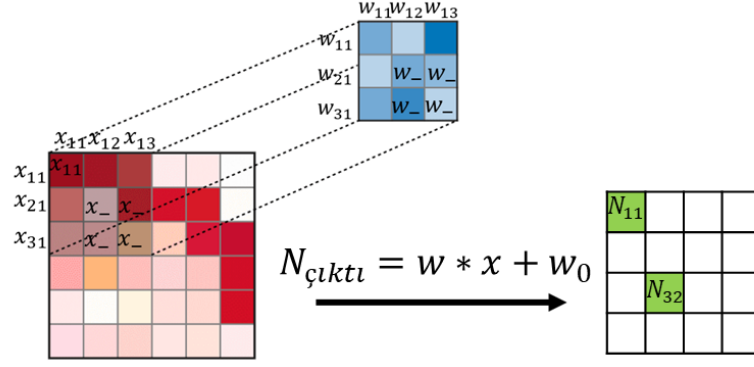
Bu bölüm devamında öncelikle ESA mimarileri için kullanılan temel katman yapıları detaylandırılmıştır. Daha sonra ESA için geliştirilmiş özel blok yapıları ve dikkat mekanizmalarına yer verilmiştir. Son olarak transfer öğrenme ve kodlayıcı ve kod çözücü yaklaşımları sunulmuştur.

2.1 Evrişimsel Sinir Ağlarının Bileşenleri

ESA mimarileri temel olarak Evrişim, Havuzlama, Aktivasyon ve Tam Bağlı katman yapılarının farklı kombinasyonlarda bir araya getirilerek oluşturulmaktadır. Bu katman yapıları aşağıda başlıklar halinde detaylandırılmıştır [76, 88].

2.1.1 Evrişim katmanı

Evrişim Katmanı, girdi verilerine evrişim işleminin uygulandığı katmandır. Evrişim işlemi temel olarak Şekil 2.2’de gösterildiği gibi bir filtreleme işlemidir. Bu filtrenin parametreleri rastgele veya belli değerler ile başlatılabilmektedir. Eğitim sırasında, bu filtrenin parametreleri, problemin çözümü için uygun öznitelikleri elde edebilecek şekilde güncellenmektedir. Şekil 2.2’de evrişim işlemi için örnek verilmiştir. Burada w filtre ağırlıkları (eğitilmesi gereken parametre), x girdi, w_0 bias ve $N_{çikti}$ evrişim çıktısını temsil etmiştir [89, 90].



Şekil 2.2: Evrişim katmanında evrişim süreci.

Şekil 2.2’de, tek kanallı ve 9×9 boyutlu görüntü matrisine 3×3 boyutlu filtresinin uygulanması ve elde edilen çıkış haritası (4×4 boyutlu) verilmiştir. Burada gösterildiği gibi w ağırlıklarına sahip 3×3 filtre (derinlik boyutu 1 için) tüm görüntüde gezdirilerek 4×4 boyutlu çıktı haritası elde edilmiştir. Şekil 2.2’de kullanılan w filtre ağırlıkları eğitim boyunca güncellenmektedir.

Evrişim katmanlarının uygulanmasında kullanılan bazı terimler şu şekildedir:

Filtre boyutu (fb): Evrişim katmanında uygulanan filtre boyutudur.

Filtre sayısı (fs): Evrişim işleminde kullanılan filtre sayısıdır.

Adım genişliği ($ad, stride$): Filtrenin gezdirme sırasında filtrenin kaydırma aralığıdır.

Dolgulama ($P, Padding$): Girdinin dış bölgelerine birler ekleyerek girdi boyutunun genişletilmesidir.

Yukarıdaki parametreleri temel alarak evrişim işlemine alınan $W_g \times H_g \times D$ (*genişlik* \times *yükseklik* \times *derinlik*) boyutundaki girdinin evrişim işleminden sonraki çıktısı Denklem 2.1’ göre hesaplanmaktadır [55, 89, 90].

$$W_{\zeta} = \frac{W_g - fb + 2P}{ad} + 1$$

$$H_{\zeta} = \frac{H_g - fb + 2P}{ad} + 1 \quad (2.1)$$

$$\text{Çıkış boyutu} = W_{\zeta} \times H_{\zeta} \times fs$$

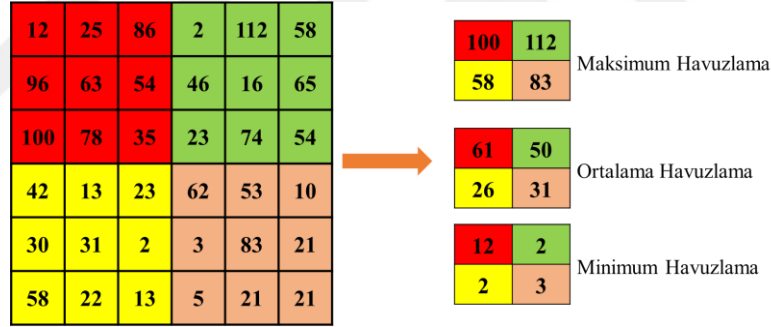
Burada W_g ve H_g sırası ile girdinin genişlik ve yükseklik değerlerini göstermiştir. Benzer şekilde W_c , ve H_c sırası ile çıktının genişlik ve yükseklik değerlerini göstermiştir. Evrişim işleminde eğitilmesi gereken parametre sayısı Denklem 2.2'ye göre hesaplanmaktadır.

$$parametreSayısı = ((fb_w \times fb_h \times fb_d) + 1) \times fs \quad (2.2)$$

Burada fb_w , fb_h ve fb_d sırası ile filtre boyutunun genişlik, yükseklik ve derinlik boyutunun değerini göstermiştir.

2.1.2 Havuzlama katmanı

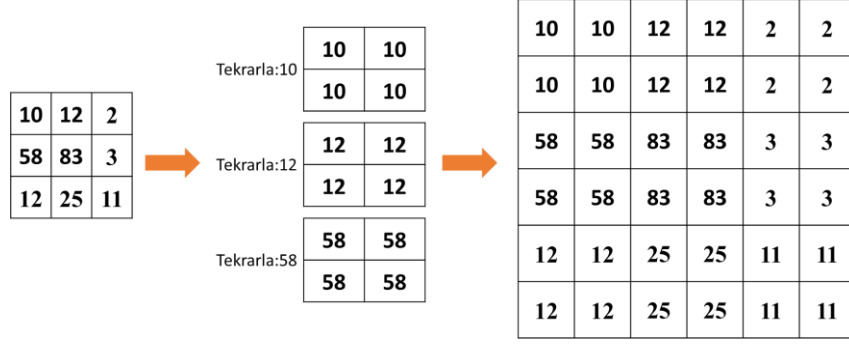
Havuzlama Katmanı, boyut küçültme katmanıdır. ESA mimarilerinde, ilerledikçe katlanarak büyük boyutlu öznetelik haritaları elde edilmektedir. Bu özneteliklerdeki anlamlı bilgileri kaybetmeden verileri küçültmek için bu katman uygulanmıştır. Bu katmanda, veriler üzerinden bir matris geçirilir ve bu matrise karşılık gelen elemanlar, ortalama havuzlama veya maksimum havuzlama gibi yöntemlerle tek bir elemana indirgenmektedir. Havuzlama katma ile ilgili örnek bir gösterim Şekil 2.3'te verilmiştir.



Şekil 2.3: Havuzlama işlemi ile ilgili örnek gösterim.

2.1.3 Yukarı örnekleme katmanı

Yukarı Örnekleme (YÖ) katmanı, havuzlama katmanının aksine girdi öznetelik haritasının boyutunu yükseltme için kullanılmıştır. Özellikle piksel seviyesinde sınıflandırma yapan UNet gibi kodlayıcı-kod çözücü (enkoder-dekoder) ağ mimarilerinde çıktı tahmin haritası, girdi görüntüsü ile aynı boyuttadır. Bundan dolayı, bu tür mimarilerde ağın sonlarına doğru boyut yükseltme katmanları ile öznetelik haritasının boyutu yükseltilmiştir. Yukarı örnekleme katmanının uygulanışı Şekil 2.4'te verilmiştir [11, 89–92].

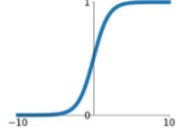
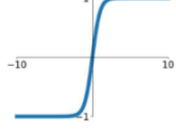
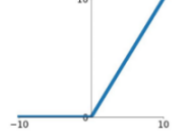
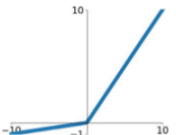


Şekil 2.4: Yukarı Örnekleme Katmanı.

2.1.4 Aktivasyon fonksiyonu

Aktivasyon katmanı, girdi verilerine doğrusal olmayan (non-linear) özellik veren katmandır. Bu sayede katmanlar arasında bağımsızlığı artırarak ESA mimarilerinin istikrarını arttığı bilinmektedir. Aktivasyon fonksiyonu için birçok farklı model önerilmiştir. Çizelge 2.1’de örnek aktivasyon fonksiyonları verilmiştir [78, 89, 90].

Çizelge 2.1: Yaygın bilinen aktivasyon fonksiyonları.

Aktivasyon Fonksiyonu	Denklemi	Grafiği
Sigmoid	$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
ReLU	$\text{relu}(x) = \max(x, 0)$	
Leaky Relu	$L_relu(x) = \max(x, 0.1x)$	

Çizelge 2.1’de görüleceği üzere ReLU aktivasyon fonksiyonu girdinin negatif değerlerini 0’a yükseltmektedir. Bu aktivasyon fonksiyonun basit yapısı sayesinde ESA mimarisinde en yaygın kullanılan modeldir. Sigmoid aktivasyon fonksiyonu girdi değerlerini 0 ile 1 arasına sıkıştıran aktivasyon fonksiyonudur. Geleneksel YSA’da yaygın kullanılmasına rağmen

sigmoid fonksiyonu, ReLU fonksiyonuna göre işlem maliyeti yüksek olduğu için ESA mimarilerinde çok tercih edilmemiştir [89, 90].

2.1.5 Tam bağlı katman

Tam Bağlı (TB) katmanın temel yapısı, geleneksel YSA ile aynıdır. Bu katmanda gelen veriler ağırlık katsayısı ile çarpılır ve toplanır (Denklem 2.3). TB katmanda kullanılan ağırlıklar eğitim boyunca güncellenmektedir [89, 90].

$$N_{\text{çıkış}} = w \times x + w_0 \quad (2.3)$$

Denklem 2.3'te w ağırlık katsayı vektörünü, x girdi vektörünü ve w_0 bias değerini göstermiştir. $N_{\text{çıkış}}$ ise tam bağlı katmanın çıkış değerini temsil etmiştir. TB genellikle ESA mimarilerinde son katmanlarında uygulanmış ve sınıflandırıcı olarak görev almıştır.

2.1.6 Giriş ve çıkış katmanı

ESA mimarilerinde giriş katmanı genellikle belli boyuttaki görüntülerdir. Bunun yanı sıra görüntülerden elde edilen farklı özelliklerde girdi olarak kullanılabilir. ESA mimarilerinde evrişim katmanının parametre sayısı ve evrişim çıktısı önceki katman ile doğrudan ilişkilidir. Bundan dolayı girdi katmanının boyutunun yüksek tutulması başarıyı arttırabilmektedir. Fakat bu durum, ESA'nın işlem maliyeti yükselecektir. Yaygın bilinen AlexNet ve VGGNet mimarilerinin girdi katmanı $224 \times 224 \times 3$ olarak ayarlanmıştır [78, 89, 90].

Çıkış katmanı ESA mimarisinin nesne sınıflandırma veya tespiti gibi ESA tahmin sonucunun verildiği katmandır. Çıkış katmanı probleme özgü olarak şekillendirilmektedir. Sınıflandırma için geliştirilen ESA'larda çıkış katmanlarında TB katmanı kullanılırken bölütleme için geliştirilen Tam Evrişimsel Ağ (TEA, Fully Convolutional Network) mimarilerinde evrişim katmanları kullanılmıştır [51, 55]. Sınıflandırma veya bölütleme gibi görevlerde kullanılan ESA mimarilerinde, çıkış katmanının sonunda sigmoid veya softmax aktivasyon fonksiyonu kullanılmıştır [11, 93].

2.1.7 Dropout ve normalizasyon katmanı

ESA mimarilerinin milyon mertebesinde nöronlara sahip olduğu için eğitim veri kümesini ezberleyebilmektedir. Bu durum aşırı öğrenme ile tanımlanır ve eğitim için oldukça yüksek performans sağlarken test için düşük sonuçlar vermiştir. ESA mimarilerinde bu durumdan

kaçınmak ve yüksek başarılar elde etmek için Dropout ve Normalizasyon katmanları kullanılmıştır [89].

Dropout katmanı (seyreltme katmanı) nöronlar arasındaki bağlantıyı rastgele olarak kesen bir yapıdır. Bu sayede ESA'da bulunan bazı nöron geçişlerinin baskın gelmesini engelleyerek farklı nöron bağlantılar ile performans artışı sağlanabilmektedir. Dropout katmanında eğitim boyunca belirlenen p ($0 < p < 1$) eşliğine göre katmanlar arasındaki bağlantılar rastgele kesilmiştir. Her adım için rasgele olarak bu süreç devam etmiştir. Test zamanında özneliklerden maksimum yarar sağlamak amacıyla katmanlar arasındaki bağlantılar kesilmemektedir. Fakat bu durumda tüm nöronlar devreye girdiği için yüksek değerler ile karşılaşmıştır. Bu durumdan kaçınmak için Dropout katmanının uygulandığı katmanlarda tüm ağırlıklar p ile çarpılmıştır [94].

Evrişim katmanlarının çıkışında elde edilen bazı öznelikler aşırı büyük veya küçük değerlerden oluşabilmektedir. Normalizasyon katmanı girdi verisindeki verileri belli bir aralıkta sıkıştırmaktadır. Bu sayede belirsiz aralıkta olan (bazı değerler oldukça yüksek) bazı öznelik haritalarının değerlerini 0 ile 1 gibi aralıklara çekilmektedir. Bu sayede ağ mimarisinin öğrenme süreci daha stabil hale getirilmiştir [88, 95].

2.2 Özel Blok Yapıları

ESA mimarileri kullanan birçok çalışmada katman sayısı artırılarak daha derin ve anlamsal öznelikler elde edilmeye çalışılmıştır. Fakat, ESA'da çok fazla katman içermesi, kaybolan gradyan problemine neden olmuştur [76, 82, 88].

2.2.1 Kaybolan gradyan problemi

ESA mimarilerinde parametre eğitim için geri yayılım modeli uygulanır. Geri yayılımda kullanılan temel model Denklem 2.4'te verilmiştir [88–90].

$$w_1 = w_1 - \alpha \frac{\partial J(W)}{\partial w_1} \quad 0 < \alpha < 1 \quad (2.4)$$

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial w_{n-1}} \cdots \frac{\partial w_2}{\partial w_1} \quad (\text{Zincir Kuralı})$$

Burada J amaç fonksiyonu ve e hatayı göstermektedir. w_1 parametresinin güncellenmesi için $\frac{\partial J}{\partial w_1}$ türevi hesaplanması gerekmektedir. $\frac{\partial J}{\partial w_1}$ hesaplamak için zincir kuralı

uygulanmıştır. $\frac{\partial e}{\partial w_{n-1}}$... zincir kurallında geçerli olan ara katmanlarının türevini temsil etmiştir. α değeri adım genişliğini göstermektedir ve genellikle küçük bir değer seçilmektedir. Bu sayede ağıın parametrelerini küçük bir miktar değiştirilerek güncellenmektedir [88].

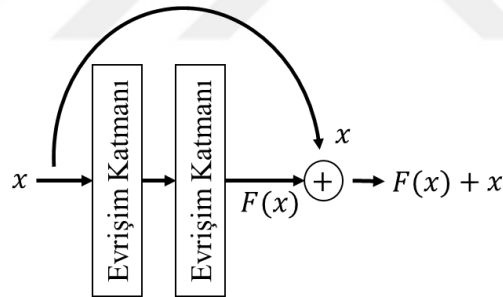
VGG19 gibi 19 katmanlı veya daha fazla katmana sahip bir mimaride zincir kuralından dolayı $\frac{\partial J}{\partial w_1}$ sifira oldukça yaklaşır. Buda ağıın ilk katmanlarının eğitilmesini engeller. Bu probleme kaybolan gradyan problemi denilmektedir.

ESA mimarilerinde kaybolan gradyan problemi ile başa çıkmak için ara bağlantı modeli geliştirilmiştir. Ara bağlantı modeli için yaygın bilinen iki versiyon vardır [82, 88, 96].

1. Eklemeli bağlantı: Artık (Residual) bağlantı [88]
2. Birleştirmeli bağlantı: Dense Blok [96]

2.2.2 Eklemeli bağlantı: Artık (residual) bağlantı

Artık bağlantı modeli temel olarak girdinin bazı katmanları atlayarak son katman çıktısına eklenmesidir [88]. Artık bağlantı ile geliştirilen Residual bloğu Şekil 2.5'te gösterilmiştir.



Şekil 2.5: Artık bağlantı modeli.

Şekil 2.5'te gösterildiği gibi artık bloklarında x girdisi öncelikle 2 evrişim katmanından geçirildikten sonra $F(x)$ çıktı elde edilmiştir. Daha sonra blok çıkışına doğru $F(x)$ çıktısı ile x değeri toplanmıştır. Bu sayede x öznelik haritası doğrudan blok çıktısı ile bağlantı sağlanmıştır. Geriye yayılım sırasında girdiden önceki katmanlarda kısa bağlantı sağladığı için önceki katmanların eğitimini kolaylaştırmaktadır [88]. Bu durum Denklem 2.5'te gösterilmiştir. Denklem 2.5'te artık bloğunun geri yayılım formülü verilmiştir [88, 89].

Normal bağlantı: $H(x) = F(x)$

$$\text{Normal bağlantı için gradient: } \frac{\partial J}{\partial x} = \frac{\partial J}{\partial H} \frac{\partial H}{\partial x} = \frac{\partial J}{\partial H} \frac{\partial F}{\partial x} \quad (2.5)$$

Artık bağlantı: $H(x) = F(x) + x$

$$\text{Artık bağlantı için gradient: } \frac{\partial J}{\partial x} = \frac{\partial J}{\partial H} \frac{\partial H}{\partial x} = \frac{\partial J}{\partial H} \left(\frac{\partial F}{\partial x} + 1 \right) = \frac{\partial J}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial J}{\partial H}$$

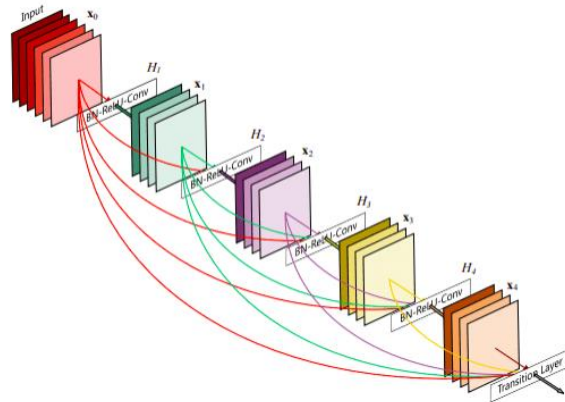
Denklem 2.5'te gösterildiği gibi girdiden önceki katmanın güncellemesi için normal bağlantıda $\frac{\partial J}{\partial H} \frac{\partial F}{\partial x}$ formülü ile ilerlerken artık bağlantıda $\frac{\partial J}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial J}{\partial H}$ şeklinde ilerlemektedir. Bu durumda $\frac{\partial J}{\partial H} \frac{\partial F}{\partial x}$ değeri sıfıra çok yakın olsa bile eklenen $\frac{\partial J}{\partial H}$ sayesinde önceki katmanın eğitilmesini mümkün kılmıştır [88].

Artık blok yapısı ile geliştirilen çalışmada [88] ResNet50 (50 katman), ResNet101 (101 katman) ve ResNet152 (152 katman) ağ mimarileri geliştirilmiş ve çok katmanlı bu ağ mimarileri birçok problem için yüksek sonuçlar sağlamıştır [88, 97, 98].

2.2.3 Birleştirmeli bağlantı: Dense blok

UNet ve DenseNet gibi ağ mimarileri önceki katmandan elde edilen öznetelikleri doğrudan son katmanlarda kullanılması için birleştirmeli bağlantı modeli kullanılmıştır. Bu sayede artık bağlantılar gibi kaybolan gradient problemine çözüm sağlanmıştır Ayrıca ilk katmanlardaki öznetelikleri son katmana taşıyarak ilk katmanlardaki özneteliklerden daha fazla yararlanmışlardır [67, 88, 96, 99].

Birleştirme modelini kullanan en ünlü sınıflandırma mimarilerinden bir tanesi DenseNet'tir [96]. DenseNet ağında Dense blokları ile geliştirilmiştir. Dense bloğu Şekil 2.6'da verilmiştir.



Şekil 2.6: Dense blok yapısı [96].

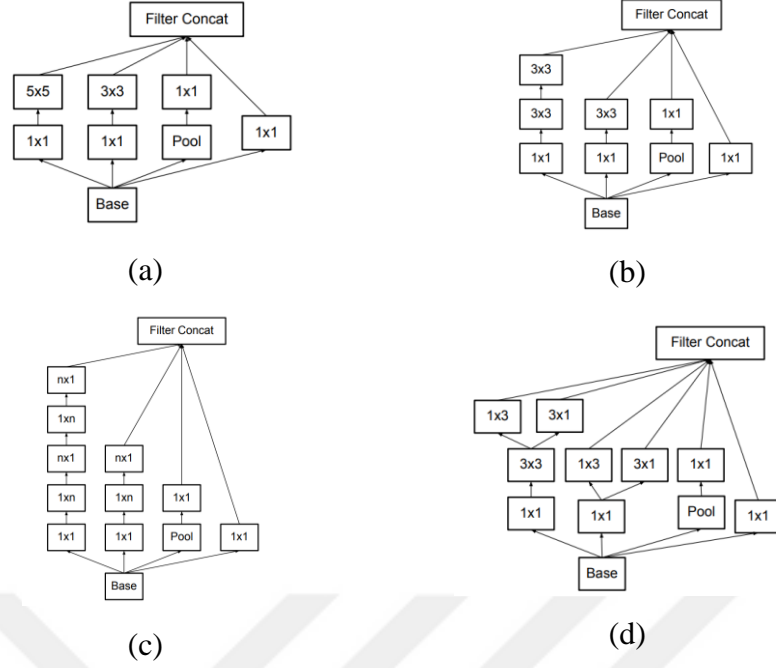
Şekil 2.6’da görüleceğe üzere dense bloğunda önceki katmanlarda elde edilen öznitelikler sonraki tüm katmanların girdisine verilmiştir. Birleştirme sırasında iki öznitelik haritası derinlik boyutunda birleştirilmiştir. Dense Blok yapısı oldukça etkili bir yapı sergilese de arka arkaya yapılan birleştirmeler oldukça büyük boyutlu bir öznitelik haritasına sebep olur. Bundan dolayı dense blok yapılarının hesaplama karmaşıklığı ve zaman tüketimi fazladır [96].

Eklemeli bağlantı ve birleştirmeli bağlantı modellerini kullanılarak birçok yeni özel blok yapıları geliştirilmiştir. Bunlar aşağıda başlıklar halinde detaylandırılmıştır.

2.2.4 Inception blok

Inception bloğu, ImageNet 2014 yarışmasında birinci olan GoogleNet [100] mimarisinde ilk olarak tanıtılmıştır. Inception bloğunun temel amacı standart evrişim katmanı kullanmak yerine bir katmanda farklı çekirdek boyutunda erişimlerinin uygulanmasıdır. Bu bakış açısıyla geliştirilen Inception blok yapısı, parçalı bir evrişim süreci uygulamak ve evrişim çıktılarını birleştirmeye dayanmaktadır. Bu sayede girdi verisinden düşük parametre ile daha güçlü öznitelikler elde edilmesi sağlanmıştır.

Orijinal Inception Blok (IB) yapısı Şekil 2.7.a’da gösterilmiştir. Burada görüleceği gibi, girdi katmanına paralel olarak farklı evrişim katman süreçleri uygulanmıştır. Daha sonra elde edilen çıktı birleştirilmiştir. GoogleNet (Inception v1) ağ yapısının üstün performansını ilham alan Szegedy ve dig. [101], Şekil 2.7’de verilen 3 farklı Inception blok versiyonu önermiştir. Bunlar sırası ile Inception bloğu 1 (IB_1), Inception bloğu 2 (IB_2) ve Inception bloğu 3 (IB_3) yapılarıdır. Bu blok yapıları InceptionV3 [101] mimarisinde kullanılmış ve yüksek başarılar sağlanmıştır.

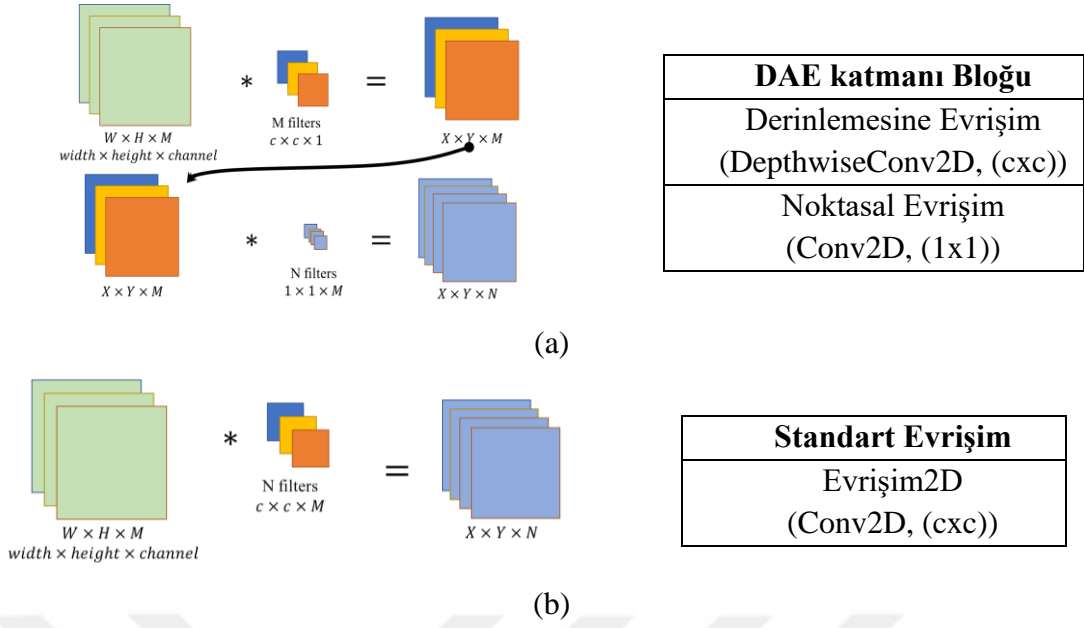


Şekil 2.7: Inception blok yapıları [101], a) Orijinal IB, b) IB1, c) IB2, d) IB3.

IB1’de (Şekil 2.7.b), orijinal IB’de kullanılan 5×5 evrişim katmanı yerine 3×3 çekirdek boyutuna sahip iki evrişim katmanının kullanımına dayanmaktadır. Öte yandan, IB2’de (Şekil 2.7.c), $n \times n$ çekirdek boyutuna sahip evrişim için sıralı $n \times 1$ ve $1 \times n$ çekirdek boyutuna sahip evrişim katmanları uygulanmıştır. IB3’te (Şekil 2.7.d) ise IB1’deki son uygulanan 3×3 evrişim yerine 3×1 ve 1×3 çekirdek boyutlarına sahip evrişim katmanları uygulanmıştır.

2.2.5 Derinlemesine ayrılabilir evrişim

Derinlemesine Ayrılabilir Evrişim (DAE, Depthwise Separable Convolution) bloğu standart evrişim katmanı yerine iki parçalı bir evrişim sürecinden oluşmaktadır [95]. Bu sayede parametre sayısını önemli ölçüde düşürdüğü görülmüştür [55, 95, 102]. DAE blok yapısı Şekil 2.8’de verilmiştir.



Şekil 2.8: Derinlemesine ayrılabilir evrişim (a) ve standart evrişim (b) yapıları.

Şekil 2.8’de gösterildiği gibi, DAE katmanı iki adımdan oluşmaktadır: Derinlemesine evrişim (DE) ve Noktasal evrişim (NE). DE katmanında temel olarak standart evrişim katmanından farklı olarak girdinin her bir kanalı için ayrı bir filtre uygulanmıştır. Dolayısıyla Şekil 2.8.a’da gösterildiği gibi $W \times H \times M$ boyutunda bir girdi için M adet filtreye ihtiyaç vardır. DE katmanı Denklem 2.6’da ki gibi formülize edilmiştir.

$$O_{DE} = \sum_i^M I^i * F_{fg:c \times c}^i \quad (2.6)$$

Burada M , girdinin (I) kanal sayısını temsil ederken i ise her bir kanalda dolaşımını sağlayan parametredir. $F_{fg:c \times c}^i$, i ’nci kanala ait ve filtre genişliği (fg) $c \times c$ olan filtreyi temsil etmiştir. O_{DE} ise DE katmanın çıktısıdır. DAE işleminin ikinci aşamasında elde edilen O_{DE} çıktısına Noktasal Evrişim (NE) işlemine alınmıştır. NE temel olarak standart evrişim ile aynıdır. Fakat burada standart olarak çekirdek boyutu 1×1 ’dir. NE katmanı Denklem 2.7’de gösterilmiştir.

$$O_{NE} = O_{DE} * F_{(1 \times 1)}^{fs:N} \quad (2.7)$$

Burada, O_{NE} noktasal evrişim işlem sonucunu temsil ederken $F_{(1 \times 1)}^{fs:N}$ ise filtre sayısı N olan ve NE evrişim işlemindeki filtreleri göstermiştir.

Denklem 2.6 ve 2.7'deki işlemler göz önüne alındığında, toplam DAE hesaplama karmaşıklığı ($cost_{DAE}$) aşağıdaki gibi hesaplanmıştır [55, 62]:

Bir filtreme için maliyet:

$$cost_{I*F} = D_f^2 \cdot D_p^2$$

DE maliyeti:

$$cost_{DE} = M \cdot cost_{I*F} = M \cdot D_f^2 \cdot D_p^2$$

Bir filtreme (çb:1x1) için maliyet:

$$cost_{O_{DE*F(1x1)}} = M \cdot D_p^2$$

(2.8)

NE Maliyeti:

$$\begin{aligned} cost_{NE} &= N \cdot cost_{O_{dw*F(1x1)}} \\ &= N \cdot M \cdot D_p^2 \end{aligned}$$

$$cost_{DAE} = cost_{DE} + cost_{NE}$$

Toplam Maliyet:

$$\begin{aligned} &= M \cdot D_f^2 \cdot D_p^2 + N \\ &\quad \cdot M \cdot D_p^2 \end{aligned}$$

Denklem 2.8'de gösterilen parametrelerinin açıklaması aşağıda verilmiştir.

*: evrişim operatörünü

F : DE'de uygulanan filtre,

D_f^2 : DE'de uygulanan filtrelerdeki eleman sayısı (Not: NE'de bu değer 1 dir.);

D_p^2 : DE ve NE filtrenin dolaşma sayısı diğer bir deyişle çıktının boyutu;

M : Girdi kanal sayısı veya DE'de uygulanan filtre sayısı;

N : NE'de uygulanan filtre sayısı;

Standart evrişimin hesaplama karmaşıklığı şu şekilde hesaplanmıştır: $N \cdot M \cdot D_p^2 \cdot D_f^2$ [55]. İki hesaplama maliyet (hesaplama karmaşıklığı) karşılaştırıldığında, sonucu $\frac{M \cdot D_f^2 \cdot D_p^2 + N \cdot M \cdot D_p^2}{N \cdot M \cdot D_p^2 \cdot D_f^2} = \frac{1}{N} + \frac{1}{D_f^2}$ elde edilmektedir. Bu sonuçlar DAE işleminin çok daha az maliyetli olduğu anlamına gelmektedir [55, 62].

2.3 Dikkat Mekanizmaları

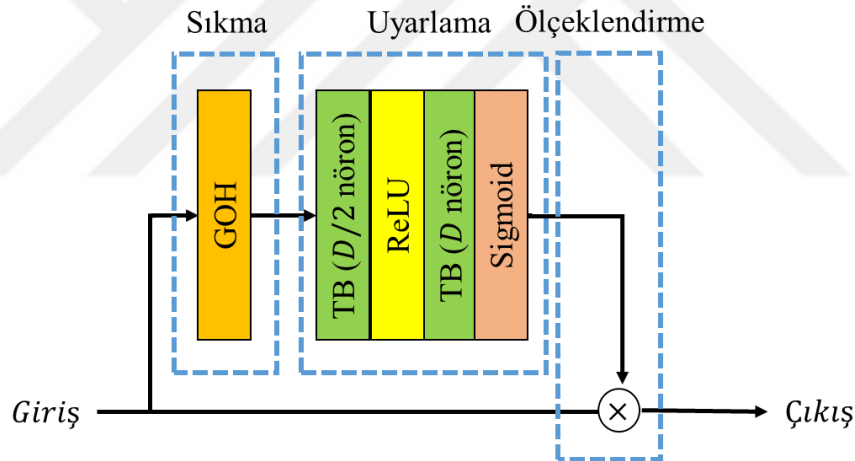
İnsan beyin yapısını ilham alarak geliştirilen ESA mimarileri örüntü sınıflandırma, nesne algılama ve anlamsal bölütleme gibi birçok alanda başarı sağlamıştır. Bununla birlikte son zamanlarda araştırmacılar insanın görme ve algılama yapısından ilham alarak dikkat mekanizmaları geliştirmişlerdir [103–105]. Dikkat mekanizmaları doğal dil işleme alanında

yaygın olarak kullanılması ile bilgisayarlı görü alanında dahil edilmiştir [105, 106]. Temel olarak bilgisayar görü alanındaki dikkat mekanizmaları ESA mimarilerinden elde edilen öznitelikler arasında önemli ayrıntıları ortaya çıkartmak için kullanılmıştır. Özellikle çok sayıda parametre içeren derin ağlarda milyon mertebesinde öznitelikler elde edilmektedir. Bu öznitelikler arasında önemli ayrıntılar ortaya çıkartılması sayesinde ağ mimarilerinin performansını arttırmıştır [103–105].

Bilgisayarlı görü alanında birçok farklı dikkat mekanizması vardır. Bu modeller arasında bazıları aşağıda alt başlıklar halinde detaylandırılmıştır.

2.3.1 Sıkma ve uyarlama bloğu

Sıkma ve Uyarlama (SU, Squeeze and Excitation) [105] bloğu, girişine verilen öznitelikleri yeniden kalibre ederek önemli öznitelikleri ortaya çıkartmaya dayalıdır. SU bloğuna dayalı geliştirilen SENet mimarisi ILSVRC 2017 yarışmasında yüksek performans sağlamıştır. SU blok yapısı Şekil 2.9’da verilmiştir [105].



Şekil 2.9: Sıkma ve uyarlama bloğunun temel yapısı.

Şekil 2.9’da gösterildiği gibi SU bloğu 3 aşamadan oluşmaktadır.

- **Sıkma:** Sıkma aşamasında girdi öznitelik boyutu küçültülmektedir. ESA mimarilerinde genellikle evrişim katmanlarından sonra 3 boyutlu ve oldukça fazla öznitelikler elde edilmektedir. Dikkat mekanizması için bu özniteliklerden özet bilgi çıkartılması gerekmektedir. Bundan dolayı sıkma biriminde havuzlama katmanları kullanılmıştır. Orijinal SU modelinin tanıtıldığı çalışmada [105] Global Ortalama Havuzlama (GOH, Global Average Pooling) katmanının kullanılması daha etkili olduğu tespit edilmiştir.

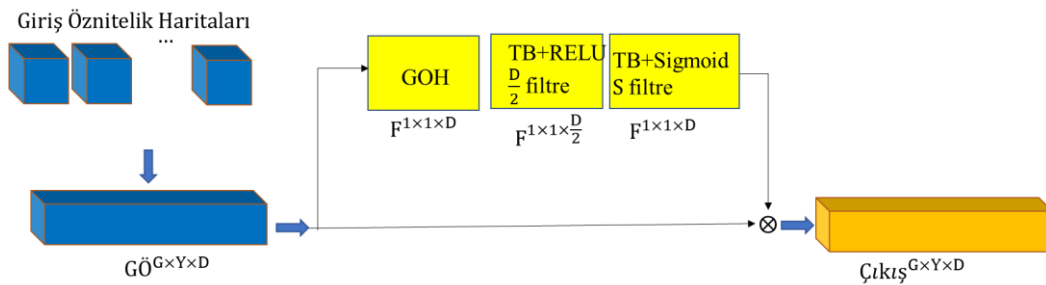
- **Uyarlama:** Diğer adıyla Yeniden Kalibrasyon (Adaptive Recalibration) bölümünde girdi öznitelik haritası için ağırlıklar ayarlanmaktadır. Uyarlama biriminde temel olarak tam bağlı katmanlar kullanılmıştır. Tam bağlı katmanlarda, sıkma biriminde elde edilen özet bilgi işlenerek ağırlıklar elde edilmiştir. Bu ağırlıklar çok farklı aralıkta olabileceği için sigmoid fonksiyonu kullanılarak ağırlıklar ölçeklendirilmiştir. Sigmoid fonksiyon çıktısında ağırlıklar 0 ile 1 arasında sıkıştırılmıştır. Bu ağırlıklarda 0 yakın önemsiz öznitelikler ve 1 yakın önemli öznitelikleri temsil etmektedir.
- **Ölçeklendirme:** Bu aşama aslında Uyarlama aşamasının bir parçasıdır. Uyarlama biriminde elde edilen öznitelikler elemanları tekrarlanarak girdi öznitelikler ile aynı boyuta getirilmiştir. Daha sonra girdi ve ağırlıklar eleman bazlı çarpma işlemi uygulanmıştır. Bu sayede girdi öznitelikleri 0 ile 1 arasında değişen ağırlıklar ile yeniden kalibre (Uyarlama) edilmiştir.

Sıkma ve Uyarlama (Squeeze and Excitation) [105] bloğunun üstün performansını ilham alan araştırmacılar tarafından Sıkma ve Uyarlama için yeni yaklaşımlar geliştirilmiştir. En çok bilenen yeni yaklaşımlar şunlardır [105, 107]:

- Mekânsal Sıkma ve Kanal Uyarlama (Spatial Squeeze and Channel Excitation),
- Kanal Sıkma ve Mekânsal Uyarlama (Channel Squeeze and Spatial Excitation),
- Mekânsal ve Kanal bazlı Sıkma ve Uyarlama Bloğu (Spatial and Channel Squeeze & Excitation Block).

2.3.1.1 Mekânsal sıkma ve kanal uyarlama

Mekânsal Sıkma ve Kanal Uyarlama (MSKU) bloğu orijinal SU modelini temsil eder. Bunun yanında blok yapısında kullanılan tam bağlı katman ve bağlantı yapısında değişiklikler yapılarak yeni modeller oluşturulmuştur [108, 109]. MSKU bloğunun detayları Şekil 2.10'da verilmiştir.

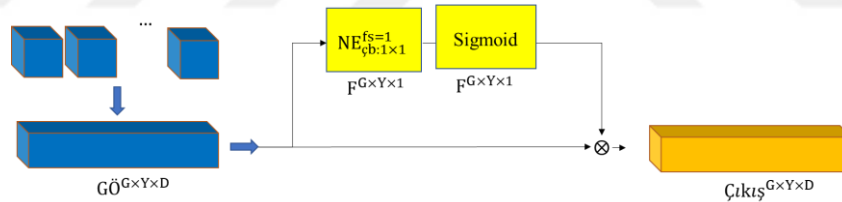


Şekil 2.10: Mekânsal Sıkma ve Kanal Uyarlama (MSKU).

Şekil 2.10’da anlaşılacağı üzere, MSKU’nun sıkma aşamasında; girdiye Global Ortalama Havuzlama (GOH, Global Average Pooling) katmanı uygulanmıştır. Bu katman sayesinde 3 boyutlu öznelik haritası yükseklik ve genişlik boyutları sıkıştırılarak bir vektöre dönüştürülmüştür. Yani her bir kanal için mekânsal bilgilerin ortalaması alınmıştır. Uyarlama aşamasında öncelikle TB (D/2 nöronlu) katmanı ve ReLU fonksiyonu uygulanmıştır. Daha sonra sırası ile TB (D nöronlu) katmanı ve sigmoid fonksiyonu uygulanmıştır. Bu sayede, girdi için bir ağırlık vektörü elde edilmiştir. Son olarak bu ağırlık vektörü girdi ile eleman bazlı çarpma işlemi uygulanmıştır [105, 108, 109].

2.3.1.2 Kanal sıkma ve mekânsal uyarlama

MSKU’da GOH katmanı uygulandığı için girdinin mekânsal (uzamsal) bilgileri (kenar ve renk vb.) tek bir noktada toplanmaktadır. Bu da konum bilgisi gerektiren bölütleme problemlerinde mekânsal bilgilerin kaybolmasına sebep olabilmektedir. Bu problemden yola çıkarak araştırmacılar tarafından Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu geliştirilmiştir. KSMU modelinin MSKU’dan farkı GOH katmanı yerine noktasal evrişim katmanı kullanmasıdır [107].



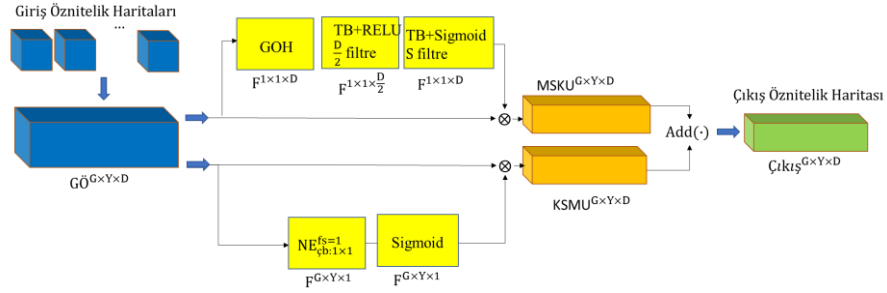
Şekil 2.11: Kanal Sıkma ve Mekânsal Uyarlama (KSMU).

Şekil 2.11’de verilen KSMU bloğunun girdisine 1 filtreli noktasal evrişim uygulanmıştır. Bu sayede öznelik haritasındaki derinlik boyutunda bulunan bilgiler bir noktada toplanmıştır. Daha sonra noktasal evrişim çıktısına sigmoid fonksiyonu uygulanarak çıktı ağırlıkları (W) elde edilmiştir. Çıktı ağırlıkları, girdinin yüksekliği ve genişliği ile aynı boyutta ve 0 ile 1 arasında değişen değerlere sahip bir matristir. Son olarak bu ağırlık matrisine girdi ile eleman bazlı çarpma işlemi uygulanmıştır [107].

2.3.1.3 Mekânsal ve kanal bazlı sıkma ve uyarlama

MSKU ve KSMU blokları farklı yapıda geliştirildiği için ikisinin önemli avantajları bulunmaktadır. Roy ve dig. [107], iki blok yapısını birleştirerek Mekânsal ve Kanal bazlı

Sıkma ve Uyarlama (MKSU, Spatial and Channel Squeeze & Excitation Block) bloğu önermişlerdir. MKSU blok yapısı Şekil 2.12’de verilmiştir.



Şekil 2.12: Mekânsal ve Kanal bazlı Sıkma ve Uyarlama (MKSU).

Şekil 2.12’de verilen MKSU bloğunda girdi öznitelik haritalarına MSKU ve KSMU blokları paralel olarak uygulanmıştır. Elde edilen MSKU ve KSMU çıktıları daha sonra birleştirilmiştir. Roy ve dig. [107], MSKU ve KSMU çıktılarını birleştirmek için 4 farklı model önermişlerdir.

- **Birleştirme:** MSKU ve KSMU çıktılarını doğrudan derinlik (kanal) boyutunda birleştirilmiştir.
- **Toplama:** MSKU ve KSMU çıktıları, eleman bazında toplanmıştır.
- **Max-Out:** MSKU ve KSMU çıkışlarının her noktası için büyük değer seçilmiştir
- **Çarpma:** MSKU ve KSMU çıktıları, eleman bazında çarpılmıştır.

2.3.2 ViT dönüştürücüsü

Dönüştürücüler (Transformers) doğal dil işleme alanında ilk olarak yayınlanmasından itibaren büyük ilgi görmüştür [106, 110]. LSTM gibi modellerin yerine geçen dönüştürücüler oldukça başarılı sonuçlar vermiştir. Dönüştürücülerin en önemli özelliği kelimeler arasındaki ilişkiyi inceleyen öz-dikkat mekanizmasına sahip olmasıdır. Öz dikkat mekanizmasında temel olarak tüm kelimeler arasındaki ilişkiyi inceler. Bu sayede girdiyi bir bütün olarak ele alır. ESA mimarilerindeki Evrişim katmanı ise filtre boyutu kadar bilgileri işler. Bu durum literatürde bölgesellik (yererlilik, locality of convolution operations) olarak tanımlanmıştır. Evrişim işlemindeki bölgesellik probleminden dolayı küresel anlamsal öznitelikleri (farklı noktalardaki bilgileri ilişkilendirerek elde edilen öznitelikler) çıkartmasını engeller. Bunun aksine dönüştürücülerde tüm girdiyi bir bütün olarak işlediği için güçlü küresel anlamsal öznitelikler (uzun menzilli bağlamsal bilgisi, long-range contextual information) elde eder [110].

Dosovitskiy ve dig. [106], 2020 yılında dönüştürücüleri bilgisayarlı görü alanına uyarlamak için Vision Transformer (ViT) modelini önermişlerdir. ViT modelinin gelişmesi ile dönüştürücüler bilgisayarlı görü alanında aktif olarak uygulanmış ve son zamanlarda oldukça yaygınlaşmıştır ViT modelinde sırası ile Parçalama, Projeksiyon, Konumsal yerleştirme ve çok başlıklı öz dikkat işlemlerinden oluşmaktadır. Bu işlemler aşağıda başlıklar halinde detaylandırılmıştır [106, 111–113].

2.3.2.1 Parçalama:

ViT modelinde parçalama işlemi temel olarak doğal dil işleme ile aynıdır. Parçalama (tokenize, tokenized) işleminde girdi (görüntü ya da girdi öznitelik haritası) parçalanarak tokenler (jetonlar) elde edilmektedir. Parçalama işleminde $W \times H \times D$ (W : genişlik, H : yükseklik, D : derinlik) boyutundaki girdi $k \times k$ parçaya bölünmüştür. Bu işlem sonucunda $k \times k \times D$ boyutunda $\frac{W}{k} \cdot \frac{H}{k} = \frac{WH}{k^2}$ adet parça elde edilmiştir. Bu parçalar daha sonra vektörleştirilerek her parça için bir öznitelik vektörleri elde edilmiştir. Her bir öznitelik vektörü $k^2 D$ boyutundadır. Bu öznitelik vektörleri birleştirilerek dönüştürücü girdi matrisi $(\frac{WH}{k^2} \times k^2 D)$ elde edilmiştir.

2.3.2.2 Projeksiyon

Projeksiyon katmanındaki amaç temel öznitelik vektörünü genişleterek daha iyi temsil etmektir. Bu sayede parçalar (öznitelik vektörleri) arasındaki ilişki daha iyi tanımlanabilir. Projeksiyon işleminde girdi matrisi (öznitelik vektörleri) tam bağlı katmandan geçirilmiştir. Uygulanan tam bağlı katman genellikle öznitelik vektörlerinin $k^2 D$ boyutundan yüksektir. Orijinal ViT modelinde arka arkaya 512 ve 1024 boyutunda tam bağlı katmanlardan geçirilmiştir [106].

2.3.2.3 Konumsal yerleştirmeler

Yukarıdaki işlemler sonucunda parçaların konum bilgisi kaybolmaktadır. Doğal dil işlemede kelimeler arasındaki konum bilgisi önemli olduğu için parçalara konum bilgisi eklenmektedir. Genellikle konum bilgisi eklerken kosinüs fonksiyonu kullanılmıştır. Bu tez çalışmasında doğal dil işleme konusuna girilmediği için detaylı bilgi için [103, 106, 114] nolu çalışmalara bakılabilir.

2.3.2.4 Öz dikkat mekanizması

Öz-dikkat mekanizmasındaki temel amaç benzerliği temel alarak parçalar aracındaki ilişkiyi incelemektir. İnceleme çıktısında parçalarının ilişkisi temel alınarak puanlanmaktadır. Öz-dikkat mekanizmasında parçalar arasındaki ilişkiler şu şekilde hesaplanmıştır [115–117]:

$$Z = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2.9)$$

Burada Q, K ve V sırasıyla sorgu (Query), anahtar (Key) ve değer (Value) matrislerini temsil etmiştir. Denklem 2.9’da verilen QK^T sorgu ve anahtar matrislerinin çarpımını göstermiştir. QK^T matris çarpımı Q ve K matrislerinde bulunan vektörlerin noktasal çarpımına denk gelmektedir. Noktasal çarpım, vektörler arasındaki benzerliği elde etmektedir. $\frac{1}{\sqrt{d}}$ değeri ile elde edilen benzerlik skoru daraltılmıştır. Daraltma işleminde sorgu veya anahtar boyutunun kareköküne (\sqrt{d}) bölünerek benzerlik puanlarını küçültülmüştür. Bu sayede çarpma operatöründen kaynaklı, değerlerin aşırı büyümesi engellenerek daha kararlı gradyanlara izin verilmiştir. Denklem 2.9’daki kullanılan $\text{softmax}(\cdot)$ fonksiyonunu ile benzerlik sonuçları 0 ile 1 arasında değişen olasılık dağılımı elde edilmektedir [115–117].

Q, K ve V değerleri girdi matrisinin W_q , W_k ve W_v matrislerinin çarpımı ile elde edilmektedir (Denklem 2.10).

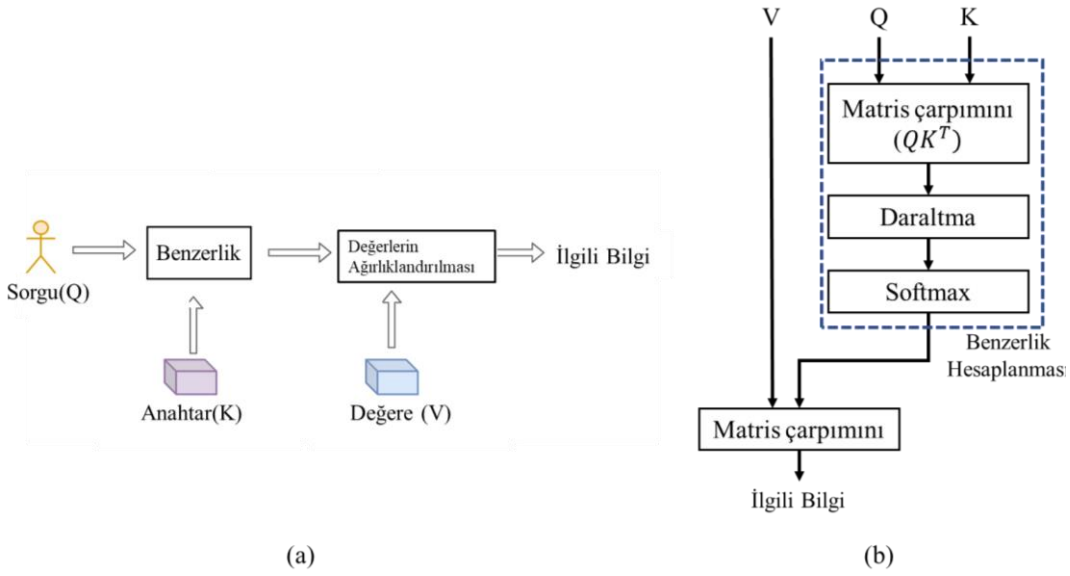
$$\begin{aligned} Q &= XW_q \\ K &= XW_k \\ V &= XW_v \end{aligned} \quad (2.10)$$

W_q , W_k ve W_v matrisleri eğitilebilir parametrelerden oluşmaktadır. Q, K ve V yapıları sırası ile şu şekilde tanımlanabilmektedir:

Q: girdi verisindeki vektörleri temsil eder. Her vektör bir kelime ya da bir görüntü parçasıdır.

K: Tüm anahtarlardır. Girdideki tüm kelimeler veya görüntü parçalarını temsil etmektedir.

V: Yine dizideki tüm kelimelerin vektör temsilidir. V çıktısı ile benzerlik skoru çarpılarak çıkış öznitelik haritası elde edilmektedir.



Şekil 2.13: Öz-dikkat mekanizmasının (a) görselleştirilmesi ve (b) hesaplanması.

Yukarıda verilen denklemlerin temel çalışma prensibi Şekil 2.13'te verilmiştir. Burada görüleceği üzere yapılacak olan q_1 sorgusu için K veri kümesinde benzerlik hesaplanmıştır. Bu benzerlik skorları V veri kümesi ile çarpılarak önemli detaylar ortaya çıkartılmış veya güçlendirilmiştir.

2.3.2.5 Çok başlıklı öz-dikkat mekanizması

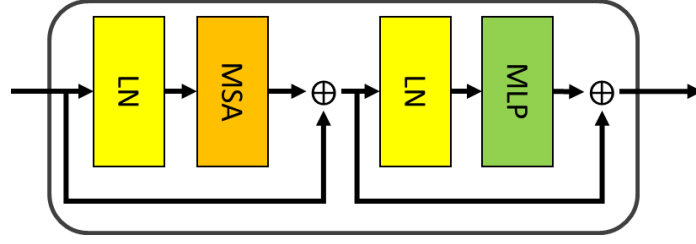
Çok başlıklı (Multihead) öz-dikkat mekanizmasında paralel olarak öz-dikkat modelini n 'defa tekrarlanmaktadır. Bu sayede daha güçlü öznitelikler elde edilme olasılığı arttırılmıştır. Tekrarlama sonucunda elde edilen Z skorları birleştirilmiştir (concat). Birleştirilen Z skoru doğal olarak yüksek boyutlu olacaktır. Z_{final} skoru elde etmek için birleştirilen Z skorları W_0 matrisi ile çarpılarak boyutu indirgenmiştir (Denklem 2.11).

$$Z_{final} = \text{MultiHead}(Q, K, V) = \text{concat}(\text{head}_1, \text{head}_2 \dots)W_0 \quad (2.11)$$

Sonuç olarak çok başlıklı öz-dikkat mekanizmasının çıktısında bir matris elde edilmiştir. Bu matrisin her bir satırı (vektörü), bir parçanın ya da kelimenin temsilidir.

2.3.2.6 ViT dönüştürücü bloğu ve tüm bileşenleri

ViT dönüştürücü bloğu, Çok Kafalı Öz-Dikkat (Multi-head Self Attention (MSA)) ve Çok Katmanlı Projeksiyon (Multi Layer Perceptron (MLP)) bloklarından oluşmaktadır (Şekil 2.14).



Şekil 2.14: Dönüştürücü mekanizması.

Şekil 2.14’te verildiği gibi, Dönüştürücü mekanizması her MSA veya MLP işleminden önce bir katman normalleştirme (LayerNorm (LN)) işlemi uygulanmıştır. Dönüştürücü bloğunun çıktısı şu şekilde hesaplanmaktadır:

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1} \quad (2.12)$$

$$z_l = MSA(LN(z'_l)) + z'_l$$

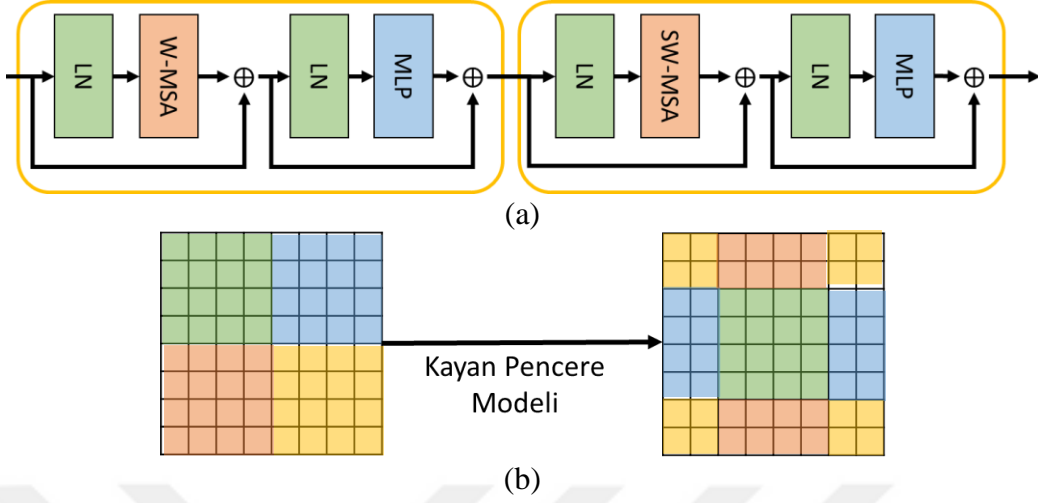
Burada z_l ve $LN(\cdot)$ sırasıyla l 'nci dönüştürücü çıkışı ve normalleştirme fonksiyonunu ifade etmiştir. z_{l-1} dönüştürücü girdisidir ve $\frac{AB}{k^2} \times C$ boyutundadır. Burada verilen $\frac{AB}{k^2}$ ve C değerleri sırasıyla elde edilen parça sayısını ve her bir parçadan elde edilen öznitelik vektörünü göstermektedir.

2.3.3 Swin dönüştürücü

ViT modelinde, ilk olarak görüntü parçalanarak tokenler elde edilmiştir. Daha sonra parçalar (token) öz-dikkat yapısından geçirilerek küresel anlamsal ayrıntılar (uzun menzilli bağlamsal bilgisi, long-range contextual information) elde edilmiştir. Bu sayede, belirli bölgeler arasındaki ilişkiyi (filtreleme) ele alan evrişim katmanlarından farklı olarak, bu yapı ile güçlü öznitelikler elde edilmiştir.

ViT dönüştürücü modeli, görüntü sınıflandırmada başarılı sonuçlar vermesine rağmen [106, 116], dönüştürücü modelinde tüm parçalar birbiri ile ilişkilendirildiği için zaman alıcı ve maliyetlidir. Ayrıca tüm parçalar birlikte ilişkilendirildiği için segmentasyon ve algılama gibi bilgisayarla görme sorunları için zayıf olduğu belirtilmiştir [111, 117, 118]. Swin dönüştürücü (Şekil 2.15) modeli, ViT modelinin hesaplama karmaşıklığını düşürmek ve bölütleme için güçlü bir dönüştürücü yapısı sergilemek için önerilmiştir [111]. Swin dönüştürücü modelinde parçalar arasındaki ilişki değerlendirilirken yerel pencere modeli kullanılmaktadır. Şekil 2.15.b’de görüldüğü gibi, Swin dönüştürücüsü, $M \times M$ parçayı

içeren pencerelerden oluşturulmuştur. Swin dönüştürücü modelinde sadece pencere içerisindeki parçalara öz-dikkat mekanizması uygulanmıştır.



Şekil 2.15: (a) Swin Dönüştürücü ve (b) pencere kaydırma modeli.

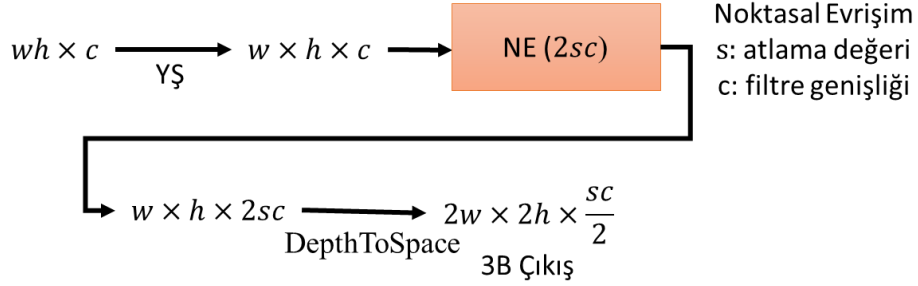
Şekil 2.15.a’da görüleceği üzere Swin dönüştürücü modeli iki adımdan oluşmaktadır. İlk adımda pencereler içerisindeki parçalar Pencere tabanlı MSA (Window based MSA (W-MSA)) modülü uygulanmıştır. İkinci aşamasında, pencereler kaydırılarak pencere içerisindeki parçalar değiştirilmiştir. Kaydırmadan sonra, kaydırılmış pencere tabanlı bir MSA (Shifted Window based MSA (SW-MSA)) modülü uygulanmıştır. Bu sayede Swin dönüştürücü modeli, ViT modelinden daha düşük bir maliyetle farklı bölgelerdeki parçalar arasındaki ilişki değerlendirilmektedir. Swin Transformer çıkış (z) Denklem 2.3’teki gibi hesaplanmaktadır [111]:

$$\begin{aligned}
 \hat{z}_l &= WMSA(LN(z_{l-1})) + z_{l-1} \\
 z_l &= MLP(LN(\hat{z}_l)) + \hat{z}_l \\
 \hat{z}_{l+1} &= SWMSA(LN(z_l)) + z_l \\
 z_{l+1} &= MLP(LN(\hat{z}_{l+1})) + \hat{z}_{l+1}
 \end{aligned} \tag{2.13}$$

Burada z_l ve z_{l+1} sırasıyla pencere tabanlı MSA ($WMSA$) ve Kaydırılmış Pencere tabanlı MSA ($SWMSA$)’nın çıktı özelliği vektörünü belirtmiştir.

2.3.4 Parça genişletme

Dönüştürücü modellerinin çıktıları vektörlermiş öznitelik haritalarıdır. Bu öznitelik haritalarının boyutunu yükseltmek için Parça Genişletme (PG) katmanı önerilmiştir [118]. Parça genişletme bloğu Şekil 2.16'da gösterilmiştir.



Şekil 2.16: Parça genişletme katmanı ayrıntıları.

PG'de ilk olarak, 2 boyutlu ($wh \times c$) girdi öznitelik haritası, $w \times h \times c$ boyuta göre yeniden şekillendirilmiştir (Reshape). Daha sonra $2 * s * c$ (s =genişletme oranı) filtreye sahip bir nokta evrişim işlemi uygulanmıştır. İşlem sonucunda $w \times h \times 2sc$ boyutunda öznitelik haritası elde edilmiştir. Son olarak, TensorFlow kitaplığında bulunan DepthToSpace (tf.nn.depth_to_space) katmanı uygulanarak bu öznitelik haritası $2w \times 2h \times \frac{sc}{2}$ boyutuna yeniden şekillendirilmiştir.

2.4 Transfer Öğrenme Yaklaşımı

Yeni bir ESA mimarisinin katman ağırlıkları genellikle rastgele atanmaktadır. Eğitim sırasında bu ağırlıklar sınıflandırma veya bölütleme için optimum değerlere güncellenmektedir. Basit bir ağ mimarisinde bile öğrenilecek yüz binlerce parametre vardır. Bu nedenle, gelişmiş bir ESA mimarisinin büyük bir veri kümesi ile eğitilmesi gerekmektedir. Ancak yüzey hata tespiti gibi bazı problemler için büyük bir veri kümesi bulmak oldukça zor olabilmektedir. Bu problem için transfer öğrenme yaklaşımı geliştirilmiştir [6, 76, 119]. Transfer öğrenme yaklaşımında, ESA mimarisinin başlangıç ağırlıkları önceden eğitilmiş bir ağ mimarisinden alınmaktadır. Daha sonra bu ağ mimarisi yeniden eğitilmektedir. Sonuç olarak belli değerler ile başlatılan ESA mimarisi istenilen probleme uygun olacak şekilde parametreleri güncellenmektedir. Bu sayede yeni bir ESA ile karşılaştırıldığında, Transfer öğrenme ile eğitilen ESA mimarisi daha yüksek performanslar üretebilmektedir [120–122].

Transfer öğrenmede genellikle bir milyondan fazla görüntü içeren ImageNet [75] gibi yüksek boyutlu bir veri kümesi ile eğitilmiş ağ mimarileri kullanılmaktadır. Bu ağ mimarileri geliştirilecek yeni mimarinin omurga ağı olarak görev alır. Omurga ağının son katmanında veya belli katmanlarından bağlantılar oluşturularak yeni ağ mimarileri geliştirilmektedir. Bu sayede önerilen ağ mimarisinin belli bir bölümü önceden eğitilmiş parametreler ile başlatılırken yeni eklenen katmanlar rastgele değerler ile başlatılmaktadır. Transfer öğrenme için kullanılan başlıca ağ modelleri Çizelge 2.1’de verilmiştir [89, 90].

Çizelge 2.1: Transfer öğrenme için kullanılan başlıca ağ modelleri.

Model	Katman Sayısı	Parametre Sayısı	ImageNet Performansı
AlexNet	8	61M	-
VGG16	16	138,4M	71,3
VGG19	19	143,7M	71,3
ResNet50	107	25,6M	74,9
ResNet101	209	44,7M	76,4
ResNet152	311	60,4M	76,6
GoogleNet-InceptionV1	27	6,8M	-
InceptionV3	189	23,9M	77,9
InceptionResNetV2	449	55,9M	80,3
MobileNet	55	4,3M	70,4
MobileNetV2	105	3,5M	71,3
DenseNet121	242	8,1M	75,0
DenseNet169	338	14,3M	76,2
DenseNet201	402	20,2M	77,3
EfficientNetB0	132	5,3M	77,1
EfficientNetB1	186	7,9M	79,1
EfficientNetB2	186	9,2M	80,1
EfficientNetB3	210	12,3M	81,6
EfficientNetB4	258	19,5M	82,9
EfficientNetB5	312	30,6M	83,6
EfficientNetB6	360	43,3M	84,0
EfficientNetB7	438	66,7M	84,3

Transfer öğrenme için ideal omurga ağı belirlemek için genellikle omurga ağının karakteristik yapısı incelenerek ve bir takım deneysel çalışma yapılarak seçilmektedir. Keras kütüphanesinde bulunan başlıca önceden eğitilmiş ağ mimarilerinin karakteristik özellikleri Çizelge 2.2’de verilmiştir [89, 90].

Çizelge 2.2: Başlıca önceden eğitilmiş ağ mimarilerinin karakteristik özellikleri [123].

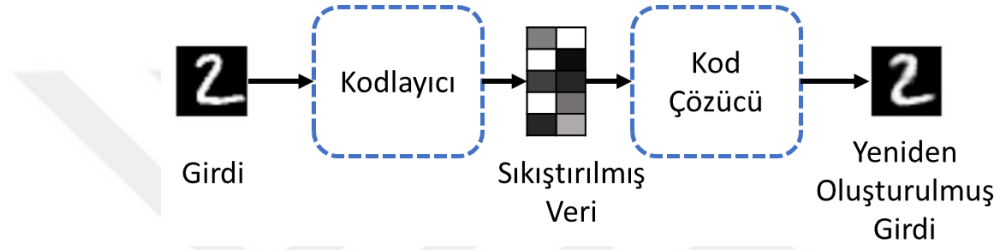
Model	Özellikler
AlexNet	<ul style="list-style-type: none">➤ ESA mimarilerinin öncüsüdür.➤ En yaygın bilinen ağ mimarisidir.➤ Geleneksel evrişim ve TB katmanlarından oluşmaktadır.
VGGNet Ailesi	<ul style="list-style-type: none">➤ Daha derin ağların daha iyi ağlar olduğu fikrine dayanarak tasarlanmıştır.➤ Eğitilmesi gereken parametre sayısı oldukça yüksektir.
ResNet Ailesi	<ul style="list-style-type: none">➤ Daha derin ağların daha iyi ağlar olduğu fikrine dayanarak tasarlanmıştır.➤ Sıfırlanan gradyan problemine çözüm olarak Residual (artık) bloklardan oluşmaktadır.➤ Katman sayısı değiştirilerek ResNet50, ResNet101 ve ResNet42 farklı versiyon geliştirilmiştir.
GoogleNet ve Inception ailesi	<ul style="list-style-type: none">➤ Inception blokları ile parametre sayısını 6,8M kadar düşürülmüştür.➤ Inception Blok yapılarını kullanılmıştır.
MobileNet Ailesi	<ul style="list-style-type: none">➤ DAE bloklarına dayalıdır.➤ Parametre sayısını düşürerek düşük hesaplama gücü olan donanımlarda kullanılması için geliştirilmiştir.
DenseNet Ailesi	<ul style="list-style-type: none">➤ Dense Blok yapılarına dayalıdır.➤ ResNet ile benzer fakat her katman, önceki katman çıktılarını ile beslenmektedir.
EfficientNet Ailesi	<ul style="list-style-type: none">➤ Derinlemesine ve noktasal evrişim katmanlarından oluşmaktadır.➤ Dense Blok yapıları içermektedir.➤ Ağ mimarilerinin amacı optimal sonuç için katman boyutlarına ve katman sayısını optimize ederek verimli ağ mimarileri oluşturmaktır.➤ Farklı katman sayısına sahip 8 versiyonu bulunmaktadır.

2.5 Kodlayıcı-Kod Çözücü Ağlar

ESA mimarileri, sınıflandırma probleminde üstün başarılar sağlamıştır. Görüntü sınıflandırma probleminin bir sonraki aşaması bölütleme ya da piksel sınıflandırmadır. Bu problemde ESA'ların görüntüdeki her pikseli ayrı ayrı sınıflandırması beklenmektedir. 2015 yılında geliştirilen UNet ağı ile bu problem için sayısız yaklaşımlar geliştirilmiştir. Bu

yaklaşımlar temel olarak kodlayıcı ve kod çözücü ağları olarak isimlendirilmiştir [7, 8, 124, 125].

Kodlayıcı ve kod çözücü ağları Şekil 2.17’de verildiği gibi kodlayıcı ve kod çözücü modüllerinden oluşmaktadır. Kodlayıcı modülünde genellikle VGGNet gibi geleneksel ESA mimarileri ile benzer yapıda oluşmaktadır. Kod çözücü modülünde ise problem için tasarlanmış evrişim, boyut yükseltme ve aktivasyon fonksiyonlarından oluşmaktadır. Bu bölümde kullanılan boyut yükseltme katmanları, kodlayıcı bölümünde veri sıkıştırmak için kullanılan havuzlama katmanı yerine kullanılmaktadır. Boyut yükseltme katmanları ile sıkıştırılmış veri genişletilmektedir [7, 67, 124].



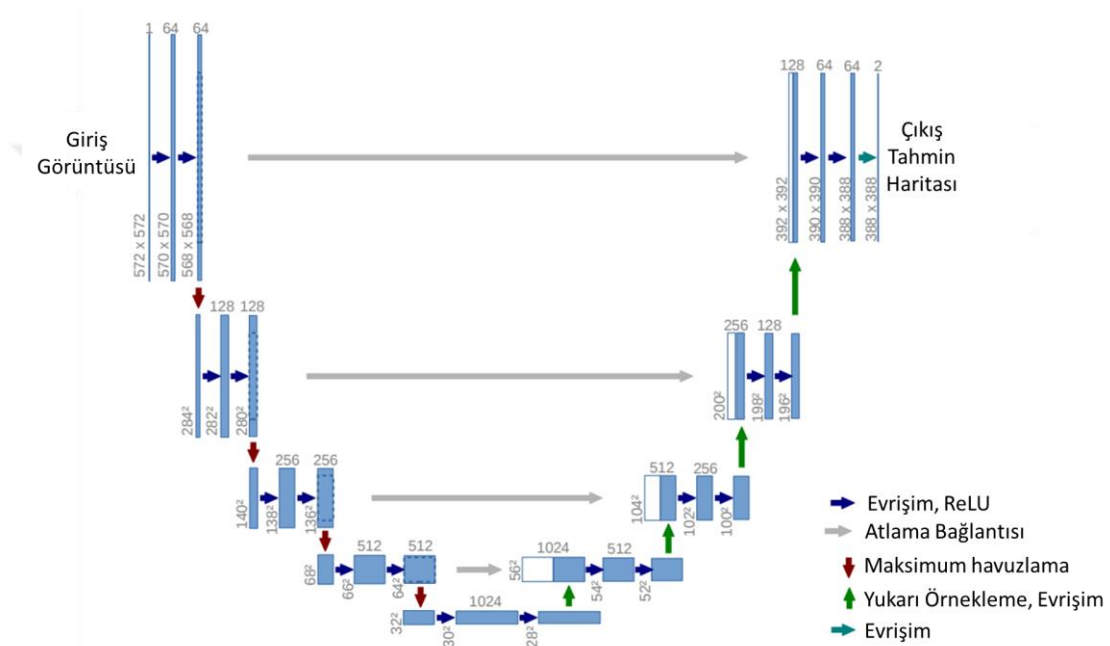
Şekil 2.17: Kodlayıcı ve kod çözücü ağlarının genel yapısı.

Şekil 2.17’de görüleceği üzere kodlayıcı modülünde girdiyi tanımlayacak sıkıştırılmış bir veri bilgisi elde edilmiştir. Kod çözücü modülünde ise kodlayıcı çıktısında sıkıştırılan veri tekrar işlenerek girdiyi temsil edecek çıktı üretilmiştir. Bu çıktı piksel seviyesinde girdinin sınıflandırması ya da girdinin farklı bir dönüşüm geçirilmiş hali olabilmektedir. Bu ağ mimarileri, AlexNet ve VGGNet gibi geleneksel ESA mimarilerinde farklı olarak ağ sonunda tam bağlı katman içermemektedir. Bu ağlarda çıktı katmanında tahmin haritası elde etmek için evrişim katmanları kullanılmıştır. Segmentasyon veya piksel sınıflandırma için kullanılan en yaygın bilinen kodlayıcı-kod çözücü mimarileri Unet [67] ve Öznitelik Piramit Ağ (ÖPA, Feature Pyramid Network) [66] mimarileridir [7, 125].

2.5.1 Unet mimarisi

U-net ilk olarak biyomedikal görüntü bölütleme için geliştirilmiştir [67]. Daha sonra gelen V-Net, FPNNet, Trans-UNet ve Swin-UNet gibi mimarinin temelini oluşturmuştur. UNet mimarisi 45 katman ve 31 milyon parametre içermektedir. Bunun yanı sıra bu mimari düşük örnekli veri kümelerinde yüksek başarı sağlamıştır. UNet mimarisi kodlayıcı, kod çözücü ve atlama bağlantılarından oluşmaktadır [67, 84, 116, 124].

Kodlayıcı birimi geleneksel ESA mimarilerine oldukça benzerdir (Şekil 2.18). Bu birimde, evrişim, aktivasyon ve havuzlama katmanlarını içermektedir. Orijinal UNet ağının kodlayıcı birimi 5 seviyeden oluşmaktadır. Her seviyede iki defa arka arkaya evrişim ve ReLU katmanları uygulanmıştır. Daha sonra (İki seviye arasında) bir havuzlama (maxpooling) katmanı uygulanarak sonraki seviyenin girdi öznelik haritası oluşturulmuştur. Uygulanan her maxpooling katmanından sonra öznelik haritası genişlik ve yükseklik boyutları yarıya indirgenmiştir. Kodlayıcı'nın son seviyesinde elde edilen öznelik haritası kod çözücü birimine aktarılmıştır (Şekil 2.18).



Şekil 2.18: Unet mimarisinin genel yapısı [67] .

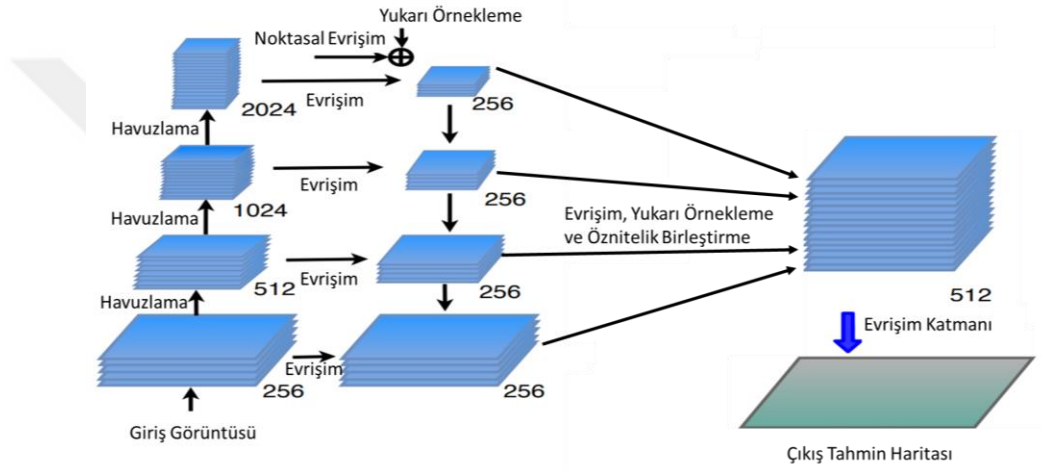
Kod çözücü birimi, temel olarak kodlayıcı birimi ile benzer bir yapıya sahiptir. Arasındaki temel fark maxpooling katmanı yerine yukarı örnekleme katmanı kullanılmasıdır. Şekil 2.18'de görüleceği üzere Unet mimarisinde kodlayıcı ve kod çözücü arasında atlama bağlantıları konulmuştur. Atlama bağlantılarının temel amacı ilk katmanlarındaki düşük seviyeli öznelikleri kod çözücü bölümüne aktarmaktır. Bu sayede piksel sınıflandırması için daha yüksek skorlar elde edilmiştir [67, 84, 116, 124].

2.5.2 Öznelik piramit ağı

Öznelik Piramit Ağ (ÖPA, Feature Pyramid Network) mimarisi 2016 yılında Lin ve diğ. [65] tarafından ilk olarak obje tespiti için geliştirilmiştir. Daha sonra önerilen model piksel

seviyesinde sınıflandırma için yeni versiyonları (Feature Pyramid Network for Multi-Class Land Segmentation) oluşturulmuştur [65, 66].

Bölütleme için geliştirilen ÖPA mimarisi [66] Şekil 2.19’da verildiği gibi iki piramit yapısına benzer bir mimariye sahiptir. Sol tarafta bulunan piramit modelin kodlayıcı birimidir. Öte yandan sağ taraftaki piramit kod çözücü birimidir. Bunun yanı sıra ÖPA mimarisinde Unet’te olduğu gibi atlama bağlantıları mevcuttur. Fakat ÖPA’da atlama bağlantılarında ekleme stratejisi uygulanmıştır. Atlama bağlantılarında ekleme stratejisi, öznetelik haritalarındaki bilgileri topladığı için daha hesaplıdır. Fakat bu durumda bazı öznetelikler kaybolabilmektedir [65, 66].



Şekil 2.19: Öznetelik piramit ağ mimarisi [65, 66].

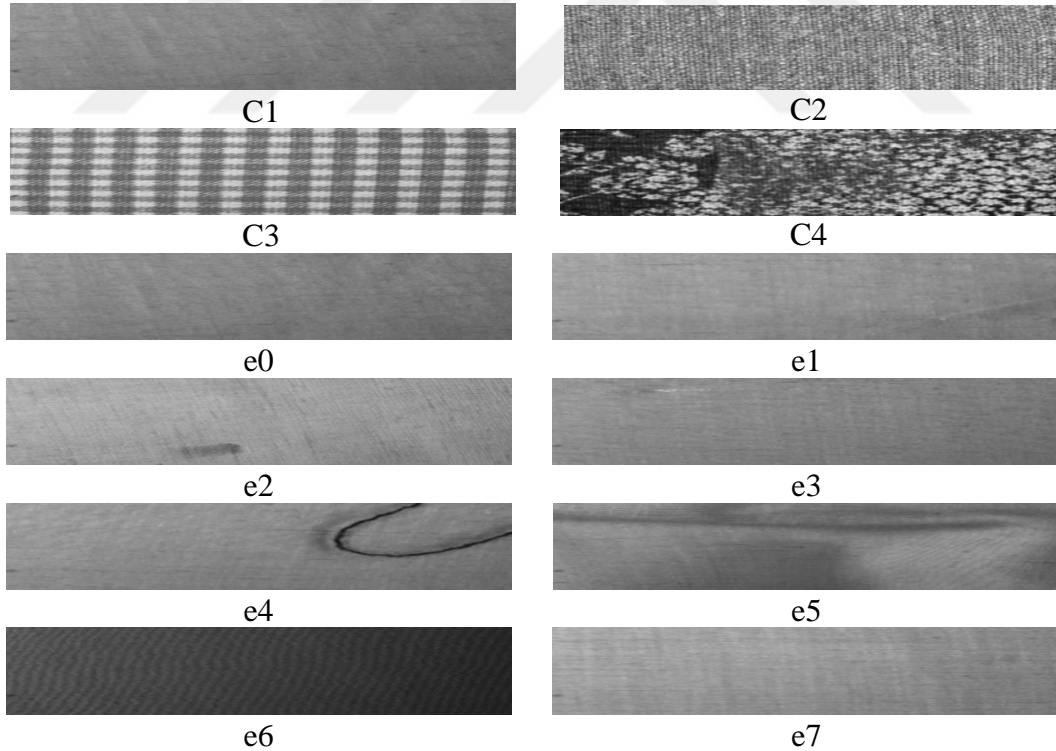
Unet’ten farklı olarak ÖPA mimarisinin ikinci önemli avantajı ağın nihai öznetelik haritası (bölütleme için doğrudan kullanılan) için kod çözücünün tüm seviyelerindeki öznetelikleri birleştirerek kullanmasıdır. Bu sayede ara katmanlardaki öznetelikleri doğrudan kullanılmasına olanak sağlamaktadır. Fakat kod çözücünün her seviyesindeki öznetelikleri son katmana taşması ve burada birleştirilmesi UNet’e göre fazladan işlem maliyetine sebep olmaktadır [65, 66].

3. YÜZEY HATA TESPİTİ İÇİN KULLANILAN VERİ KÜMELERİ

Bu tez çalışmasında, önerilen yöntemlerin sınıflandırma performanslarını analiz etmek için literatürde yüzey/doku hata tespiti için kullanılan TILDA, MT, MVTEc, MVTEc-Doku, DAGM ve AITEX veri kümeleri kullanılmıştır. Bu veri kümeleri ile ilgili detaylar alt başlıklar halinde verilmiştir. Ayrıca bölüm sonunda deneysel çalışmalarda kullanılan performans metrikleri detaylandırılmıştır.

3.1 TILDA Veri Kümesi

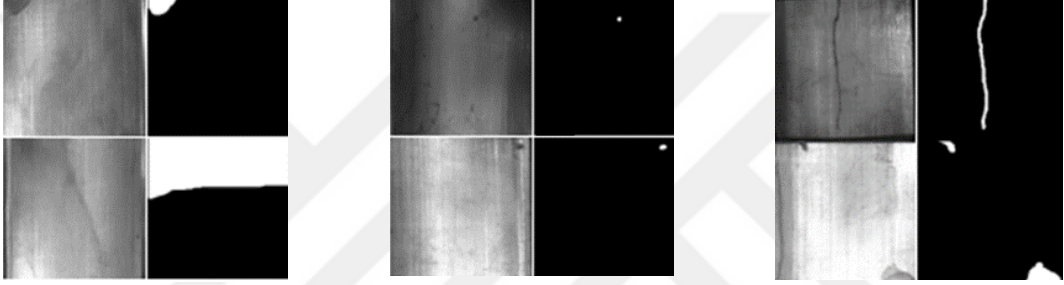
Tez kapsamında kullanılan ilk veri kümesi Computer Vision Group tarafından 1996 yılında oluşturulan TILDA [126] veri kümesidir. TILDA veri kümesi, dört ana grup (C1-C4) ve her ana grup altında iki farklı alt gruptan olmak üzere sekiz farklı kumaş türü içeren bir tekstil veri kümesidir. Her tür için 50 hatasız (e0) görüntü ve 350 hatalı (e1-e7) görüntü vardır. Görüntülerin boyutu 768×512 boyutundadır. TILDA veri kümesinden alınan örnek görüntüler Şekil 3.1’de verilmiştir. TILDA veri kümesi 1996 yılında yayınlanmasına rağmen birçok çalışmada hala aktif olarak kullanılmaktadır [37, 38, 45].



Şekil 3.1: TILDA veri kümesi (C1-C4 kumaş tipi ve e0-e6 hata sınıfları).

3.2 MT Veri Kümesi

Tez kapsamında kullanılan ikinci veri kümesi MT (Magnetic-Tile, Manyetik Karo) [69] veri kümesidir. MT veri kümesi, 1344 görüntü ve bu görüntülerin piksel seviyesindeki etiketlerini içeren bir yüzey hata tespit veri kümesidir. MT veri kümesinde 392 hatalı yüzey görüntüsü ve 952 hatasız yüzey görüntüsü içermektedir. Veri kümesinde düzensiz, yıpranma, çatlak, hava deliği ve kırılma gibi farklı yüzey hatalarını içermektedir. Ayrıca, yüzey hata tespitinde gözlemlenen ve imalat sanayi sistemlerinde karşılaşılan aşırı benzerlik, düşük kontrast ve hata boyutlarındaki değişkenlik gibi zorlukları da kapsamaktadır. Bu yönüyle hata tespiti alanında yaygın olarak kullanılan bir veri kümesidir. Şekil 3.2’de MT’den alınan örnek görüntüler verilmiştir.



Şekil 3.2: MT veri kümesinden alınmış örnek görüntüler.

3.3 MVTec Veri Kümesi

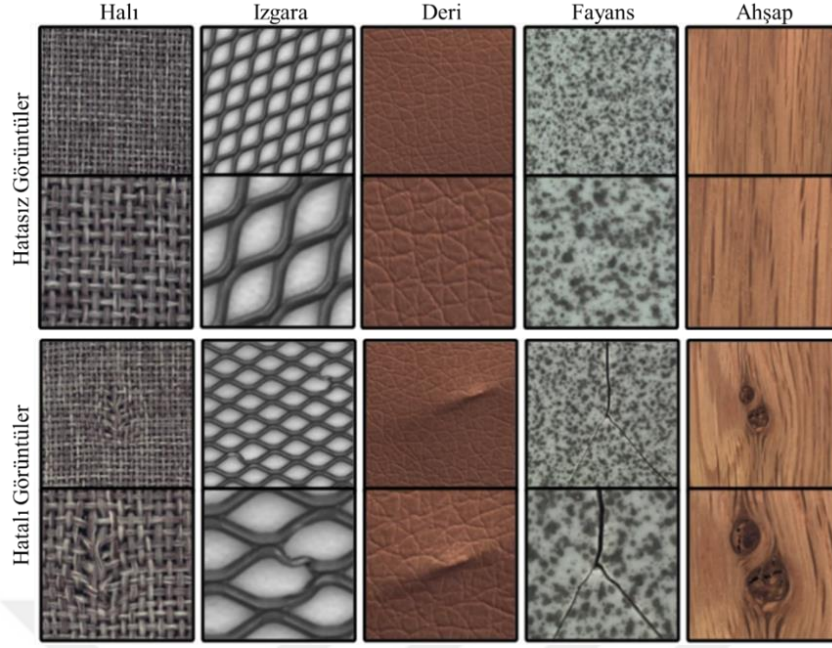
Tez kapsamında kullanılan üçüncü veri kümesi MVTec [127] veri kümesidir. MVTec veri kümesi yüzey hatası ve anormallik tespiti için kullanılan en kapsamlı veri kümelerinden biridir. Bu veri kümesinde 15 farklı görüntü sınıfını içermektedir. Bu veri kümesinde 10 sınıfı nohut, şişe ve vida gibi nesne sınıflarından oluşurken, diğer 5 tanesi halı ve deri gibi doku yüzey sınıflarından oluşmaktadır. Her sınıf için 30 ile 141 arası hatalı ve 72 ile 431 arası hatasız görüntü içermektedir. MVTec veri kümesindeki sınıflar ve sınıftaki görüntü sayısı Çizelge 3.1’de verilmiştir. Her bir sınıfın içerdiği hatalı görüntü sayısı az olması ve sınıf içerisinde karşılaşılan hata tiplerinin çok çeşitli olması MVTec veri kümesini zorlu bir veri kümesi yapmıştır. Bu nedenle genellikle otomatik kodlayıcı tabanlı yöntemlerde [73, 128–131] kullanılmıştır. Otomatik kodlayıcı modeller ile elde edilen ortalama başarı %60 F1-scorunun altında kalmıştır.

Çizelge 3.1: MVTec veri kümesinde bulunan sınıflar ve sınıflarda bulunan örnek sayıları [127].

Sınıf	Hatalı	Hatasız	Toplam
Halı	89	308	397
Izgara	57	285	342
Deri	92	277	369
Fayans	84	263	347
Odun (Ahşap)	60	266	326
Şişe	63	229	292
Kablo	92	282	374
Kapsül	109	242	351
Fındık	70	431	501
Metal somun	93	242	335
Hap	141	293	434
Vida	119	361	480
Diş fırçası	30	72	102
Transistör	40	273	313
Fermuar	119	272	391

3.4 MVTec-Doku Veri Kümesi

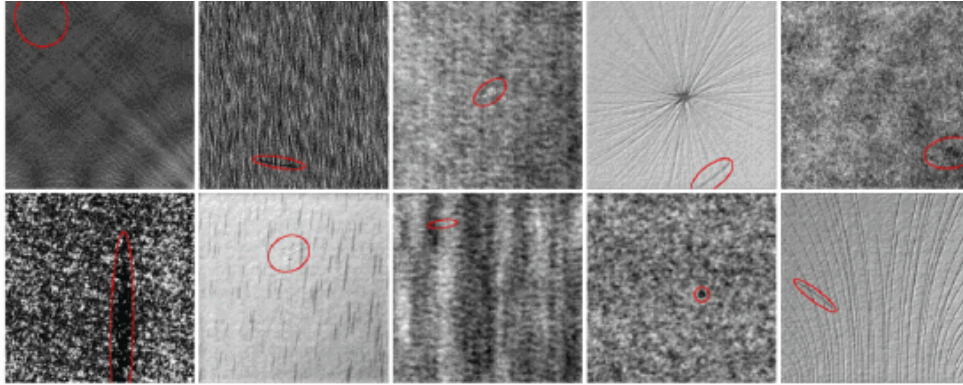
Yüzey hata tespiti için geliştirilen veri kümelerinde genellikle belirli bir yüzey sınıfına ait yüzey görüntüleri toplanmıştır. Bu tez çalışmasında yukardaki veri kümelerinde farklı olarak çeşitli doku sınıfları içeren bir veri kümesi elde etmek için MVTec-Doku veri kümesi oluşturulmuştur. Bu veri kümesi MVTec veri kümesinde bulunan Halı, Izgara, Deri, Fayans, Ahşap olmak üzere beş yüzey hata sınıfı birleştirilerek oluşturulmuştur. Şekil 3.3'te gösterildiği gibi bu veri kümesi içerdiği sınıf ve hata tipleri açısından zengindir. Ayrıca MVTec-Doku veri kümesi, hata boyutları değişken ve arka plana aşırı benzerlik gibi yüzey hata tespit zorlukları içermektedir.



Şekil 3.3: MVTec-Doku veri kümesi [127]

3.5 DAGM Veri Kümesi

Tez kapsamında kullanılan beşinci veri kümesi DAGM [132] veri kümesidir. DAGM veri kümesi, yüzey/doku hatası algılama problemi için geliştirilmiş yapay bir veri kümesidir. Bu veri kümesinde 12 farklı yüzey sınıfı bulunmaktadır. Ancak yüzey hatasına karşılık gelen etiketleme piksel seviyesinde yapılmamıştır. Bunun yerine hatalı bölgeler kabaca bir elips şeklinde oluşturulmuştur (Şekil 3.4).

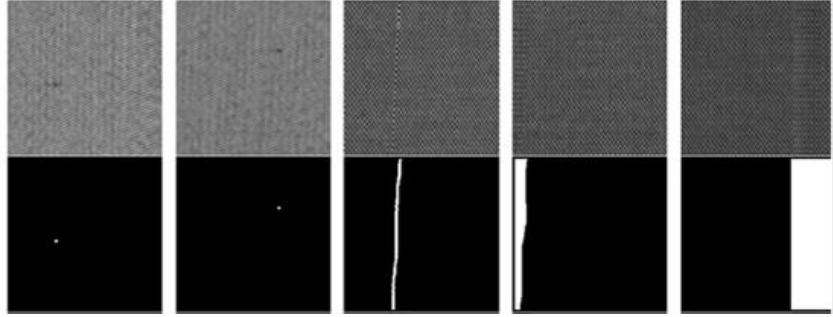


Şekil 3.4: DAGM veri kümesindeki bazı sınıflar ve elips şeklindeki etiketlemeleri [132].

3.6 AITEX Veri Kümesi

Tez kapsamında kullanılan altıncı veri kümesi AITEX [133] veri kümesidir. AITEX veri kümesi, kumaş yüzeylerinde oluşan 12 farklı hata tipinden oluşmaktadır. Bu veri kümesi

toplam 245 görüntü içermektedir. Bu görüntülerin 105 adeti kumaş hatası içermektedir. AITEX veri kümesinde bulunan görüntüler yüksek çözünürlüklüdür. Bu veri kümesinde, görüntüler 256x256 piksele bölünerek örnek sayısı arttırılmıştır. Böylece toplam 163 hatalı ve 326 hatasız görüntü elde edilmiştir.



Şekil 3.5: AITEX veri kümesinden alınmış örnek görüntüler.

3.7 Kullanılan Performans Ölçütleri

Bu tez çalışmasında, Python platformunda TensorFlow ve Keras kütüphaneleri kullanarak önerilen modeller geliştirilmiştir. Ayrıca deneysel çalışmalarda önerilen model performansını değerlendirmek için mIoU (mean Intersection over Union, bileşim üzerinde kesişim ortalaması), F1-skor, Kesinlik, Duyarlılık ve Doğruluk metrikleri kullanılmıştır. Denklem 3.1’de bu performans ölçütlerinin matematiksel ifadelerini sunulmuştur [84, 87].

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN}$$

$$\text{Duyarlılık (Recall)} = \frac{DP}{DP + YN}$$

$$\text{Kesinlik (Precision)} = \frac{DP}{DP + YP} \quad 3.1$$

$$\text{F1 - Skor} = \frac{2 \times \text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}}$$

$$\text{mIoU} = \frac{TH \cap TG}{TH \cup TG}$$

Burada DP doğru sınıflandırılmış pozitif örnekler, DN doğru sınıflandırılmış negatif örnekler, YP yanlış sınıflandırılmış pozitif örnekler ve YN ise yanlış sınıflandırılmış negatif örneklerdir. TG yüzey hata konumunu gösteren bölgeyi ve TH model tarafından üretilen tahmin bölgesini göstermektedir.

4. YÜZEY HATA TESPİTİ İÇİN ÖNERİLEN YENİ YAKLAŞIMLARI

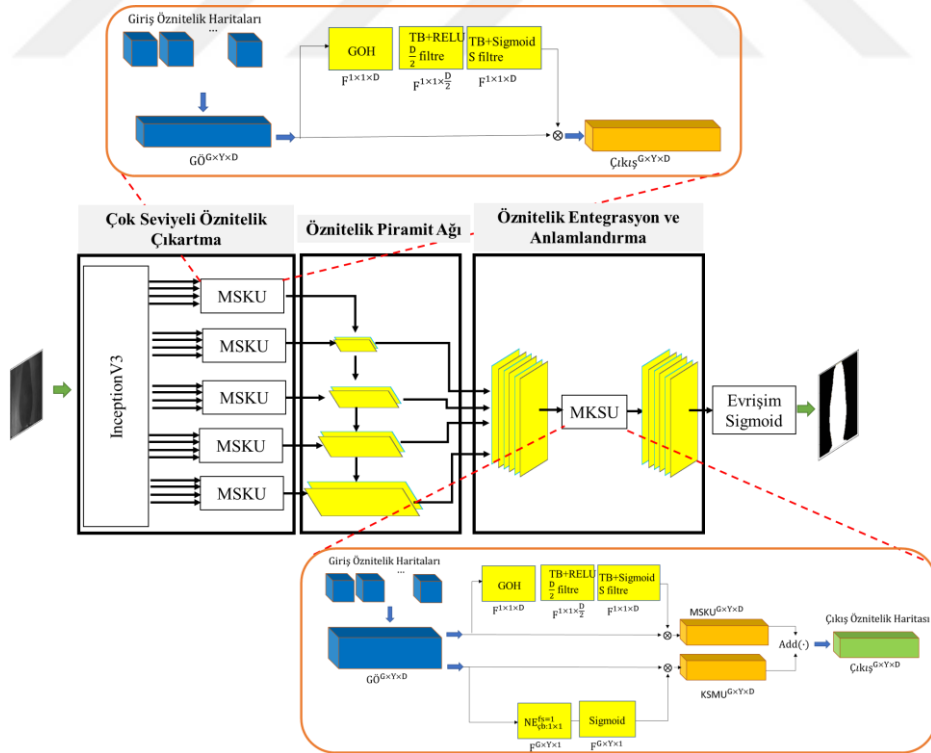
Bu bölümde, yüzey hata tespiti için önerilen derin öğrenme tabanlı yeni yaklaşımlar ile deneysel sonuçları yer almaktadır. Yüzey hata tespiti ile ilgili gerçekleştirilen çalışmalarda, Bölüm 3’te bahsedilen veri kümeleri kullanılmıştır.

4.1 Uygulama 1: Inceptionv3 Tabanlı Zenginleştirilmiş Öznitelik Entegrasyon Ağ Mimarisi

Bu uygulamada, piksel seviyesinde hata tespiti için InceptionV3 tabanlı Zenginleştirilmiş Öznitelik Entegrasyon Ağ (Inc-ZÖEA) mimarisi geliştirilmiştir. Önerilen bu mimarinin temel amacı, önceden eğitilmiş ağ mimarisinden olabildiğince fazla öznitelik çıkartılması ve MSKU ve MKSU blokları ile bu özniteliklerden anlamlı bilgilerin güçlendirilmesidir.

4.1.1 Önerilen yöntem

Yüzey hata tespiti için önerilen Inc-ZÖEA mimarisi Şekil 4.1’de verilmiştir. Önerilen model, Çok Seviyeli Öznitelik Çıkartma (ÇSÖÇ), Öznitelik Piramit Ağı (ÖPA) ve Öznitelik Entegrasyon ve Anlamlandırma (ÖEA) olmak üzere 3 modülden oluşmaktadır.



Şekil 4.1: Önerilen Inc-ZÖEA mimarisi.

Önerilen Inc-ZÖEA mimarisinin akışı Algoritma 4.1’de verilmiştir

Algoritma 4.1: Inc-ZÖEA mimarisinin akışı

Adım	İşlem	Amaç
1:	InceptionV3 ağının I_{0-19} katman çıktıları alınır.	Önceden eğitilmiş InceptionV3 ağından detaylı öznetelik haritaları elde etmek.
2:	Aynı yükseklik ve genişliğe sahip öznetelik haritaları birleştirilme ve MSKU blok uygulanması	Öznetelikler birleştirilerek zengin 5 farklı öznetelik haritası elde etmek. Bunun yanı sıra öznetelik haritalarındaki önemli ayrıntılar orta çıkartmak.
3:	Öznetelik Piramit ağına dayalı kod çözücü uygulanması	Önceki adımda elde edilen $F_{1,2,3,4,5}$ öznetelik haritalarından yeni öznetelik haritası elde etmek.
4:	Öznetelik Entegrasyon ve Anlamlandırma ağı uygulanması	Önceki adımda elde edilen $H_{l=1,2,3,4}$ öznetelik haritaları birleştirilerek nihai öznetelik haritası elde etmek.
5:	Evrişim katmanı ve sigmoid aktivasyon katmanlarının uygulanması	Önceki adımda elde edilen nihai M öznetelik haritası kullanılarak hata tespit tahmin haritası elde etmek.

Önerilen Inc-ZÖEA mimarisinin detayları alt başlıklar halinde aşağıda verilmiştir.

4.1.1.1 Çok seviyeli öznetelik çıkartma modülü

Inc-ZÖEA ağının Çok Seviyeli Öznetelik Çıkartma (ÇSÖÇ) modülünde, InceptionV3 [52] ağı kullanılmıştır. Bu bölümde InceptionV3 [52] ağından zengin öznetelik haritaları elde edilmesi amaçlanmıştır. InceptionV3 ağ mimarisinin detayları Çizelge 4.1’de verilmiştir. Burada verilen EBR, sırası ile Evrişim, Batch normalizasyon, ReLU aktivasyon katman bloğunu temsil etmektedir. Bunun yanı sıra verilen IBx , Bölüm 2.2’de detaylandırılan Inception bloklarının IB1, IB2 ve IB3 versiyonlarını temsil etmektedir.

Çizelge 4.1: InceptionV3 mimarisinin katman detayları.

Katman	Katman işlemi	Filtre sayısı	Adım genişliği	Çekirdek boyutu	Çıkış öznitelik boyutu
I_0	EBR	32	2	3x3	$96 \times 96 \times 32$
I_1	EBR	32	1	3x3	$96 \times 96 \times 32$
I_2	EBR	64	1	3x3	$48 \times 48 \times 64$
I_3	Havuzlama	-	2	3x3	$48 \times 48 \times 64$
I_4	EBR	80	1	1x1	$48 \times 48 \times 80$
I_5	EBR	192	1	3x3	$48 \times 48 \times 192$
I_6	Havuzlama	-	2	3x3	$24 \times 24 \times 192$
I_7	Mixed0		IB1		$24 \times 24 \times 256$
I_8	Mixed1		IB1		$24 \times 24 \times 288$
I_9	Mixed2		IB1		$24 \times 24 \times 192$
I_{10}	Mixed3		IB1		$12 \times 12 \times 768$
I_{11}	Mixed4		IB2		$12 \times 12 \times 768$
I_{12}	Mixed5		IB2		$12 \times 12 \times 768$
I_{13}	Mixed6		IB2		$12 \times 12 \times 768$
I_{14}	Mixed7		IB2		$12 \times 12 \times 768$
I_{15}	Mixed8		IB2		$6 \times 6 \times 1280$
I_{16}	Mixed9_0		IB3		$6 \times 6 \times 768$
I_{17}	Mixed9_1			$6 \times 6 \times 2048$	
I_{18}	Mixed10_0		IB3		$6 \times 6 \times 768$
I_{19}	Mixed10_1			$6 \times 6 \times 2048$	

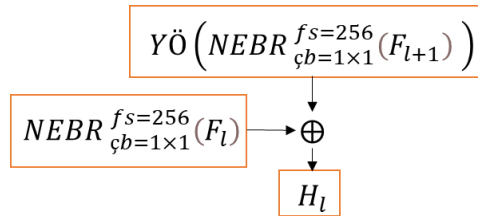
Çizelge 4.1’de verildiği gibi InceptionV3’te yer alan öznitelik haritaları (yükseklik×genişlik) 96×96 , 48×48 , 24×24 , 12×12 ve 6×6 olmak üzere 5 farklı boyuttadır. ÇSÖÇ modülünde InceptionV3 ağının I_0 – I_{19} seviyesindeki öznitelik haritaları alınmıştır. Alınan bu öznitelik haritaları Çizelge 4.2’de verildiği gibi yükseklik ve genişlik değerleri aynı olanlar birleştirilmiştir. Daha sonra birleştirilen öznitelik haritalarında bulunan önemli ayrıntıların ortaya çıkartılması için MSKU [107] uygulanmıştır. MSKU bloğunun yapısı ve detayları Bölüm 2.3’te verilmiştir.

Çizelge 4.2: ÇSÖÇ modülün elde edilen 5 farklı öznitelik haritası ve çıkış boyutları.

Çıkış Öznitelik Haritası	MSKU blok girişi öznitelik haritaları	Çıkış öznitelik harita boyutu
F_1	MSKU(I_0, I_1)	$96 \times 96 \times 192$
F_2	MSKU(I_2, I_3, I_4, I_5)	$48 \times 48 \times 560$
F_3	MSKU(I_6, I_7, I_8, I_9)	$24 \times 24 \times 1408$
F_4	MSKU($I_{10}, I_{11}, I_{12}, I_{13}, I_{14}$)	$12 \times 12 \times 7104$
F_5	MSKU($I_{15}, I_{16}, I_{17}, I_{18}, I_{19}$)	$6 \times 6 \times 9792$

4.1.1.2 Öznitelik piramit ağ tabanlı kod çözücü modülü

ÇSÖÇ modülünün sonunda 5 farklı ($F_{1,2,3,4,5}$) öznitelik haritası elde edilmiştir. Bu özniteliklerin düşük seviyeli ($F_{1,2,\dots}$) öznitelik haritalarında renk, doku ve kenar gibi mekânsal bilgiler içerirken yüksek seviyeli ($F_{\dots,4,5}$) öznitelik haritalarında anlamsal bilgiler içermektedir. Önerilen modelin kod çözücü bölümünde Öznitelik Piramit Ağ (ÖPA, Feature Pyramid Network) [66] mimarisinin kod çözücü modülünden esinlenerek bir kod çözücü modülü geliştirilmiştir. Bu modülde atlama bağlantıları için Şekil 4.2’de detaylandırılan eklemeli atlama bağlantı modeli kullanılmıştır. Buradaki F_l kod çözücü biriminde hiyerarşik olarak ilerleyen öznitelik haritası ve F_{l+1} ise atlama bağlantısı ile aktarılan öznitelik haritasıdır.



Şekil 4.2: Eklemeli atlama bağlantılarındaki öznitelik aktarma işlemi.

Şekil 4.2’de gösterildiği gibi F_l ve F_{l+1} öznitelik haritalarına sırası ile bir noktasal evrişim, batch normalizasyon ve ReLU katmanları ($NEBR_{\text{çekirdek boyutu}=1 \times 1}^{\text{filtre sayısı}=256}(\cdot)$) uygulanmıştır. Daha sonra, $NEBR_{1 \times 1}^{256}(F_{l+1})$ öznitelik haritasına yukarı örnekleme ($YÖ(\cdot)$) katmanı uygulanarak iki öznitelik haritası aynı boyuta getirilmiştir. Son olarak iki öznitelik haritasına eleman bazlı toplama işlemi gerçekleştirilmiştir. Önerilen ağın kod çözücü çıktısında her seviye için $H_{l \in \{1,2,3,4\}}$ öznitelik haritası elde edilmiştir.

4.1.1.3 Öznitelik entegrasyon ve anlamlandırma modülü

Önerilen Inc-ZÖEA ağının kod çözücü çıktısında 4 farklı $H_{l=1,2,3,4}$ öznitelik haritası elde edilmiştir. Öznitelik Entegrasyon ve Anlamlandırma (ÖEA) modülü bu öznitelik haritalarını birleştirmek ve önemli ayrıntıların ortaya çıkartılması için geliştirilmiştir. Bu modülün formülasyonu Denklem 4.1’de verilmiştir.

$$K = \begin{bmatrix} EBR_{3 \times 3}^{fs=128}(H_1) \\ YÖ^{ag=2}(EBR_{3 \times 3}^{fs=128}(H_2)) \\ YÖ^{ag=4}(EBR_{3 \times 3}^{fs=128}(H_3)) \\ YÖ^{ag=8}(EBR_{3 \times 3}^{fs=128}(H_4)) \end{bmatrix} \quad (4.1)$$

Burada öncelikle her bir H_l öznitelik haritasına sırası ile evrişim, batch normalizasyon ve ReLU katmanları ($EBR_{\text{çekirdek boyutu}=3 \times 3}^{\text{filtre sayısı}(fs)=128}(\cdot)$) uygulanmıştır. Daha sonra yukarı örnekleme ($YÖ^{\text{adım genişliği}(ag)}$) katmanı ile öznitelik haritaları aynı boyuta getirilmiştir. Bu işlemler sonunda tüm öznitelik haritaları $\frac{W}{2} \times \frac{H}{2} \times 128$ boyutuna getirilmiştir. Son olarak aynı boyuta getirilmiş öznitelik haritaları birleştirilmiştir. Bu işlem sonucunda $\frac{W}{2} \times \frac{H}{2} \times 512$ boyutunda bir öznitelik haritası elde edilmiştir. Bu öznitelik haritasındaki önemli bilgileri ortaya çıkartmak için Bölüm 2.2’de detaylandırılan MKSU [107] bloğu uygulanmıştır.

Inc-ZÖEA mimarisinin son katmanında hata tespit sonucunu elde etmek için Denklem 4.2’de gösterilen işlemler uygulanmıştır.

$$P = \sigma(E_{3 \times 3}^{fs=1}(YÖ^{s=2}(EBR_{3 \times 3}^{fs=128}(M))) \quad (4.2)$$

Burada $P_{m \times n}$ hata tespit tahmin haritası ve M , MKSU’nun çıktı öznitelik haritasını göstermektedir. σ ve $E_{3 \times 3}^{fs=1}$ sırası ile sigmoid aktivasyon fonksiyonu ve 3×3 çekirdek boyutuna sahip ve filtre sayısı $fs = 1$ olan evrişim işlemidir.

Inc-ZÖEA ağ mimarisinin eğitiminde İkili Çapraz Entropi (Binary Cross-Entropy) kayıp fonksiyonu kullanılmıştır. İkili Çapraz Entropi kayıp fonksiyonu Denklem 4.3’teki gibi tanımlanmaktadır.

$$L = - \sum_{i,j}^M y_{i \times j} \log(P_{i \times j}) + (1 - y_{i \times j}) \log(1 - P_{i \times j}) \quad (4.3)$$

Burada $y_{i \times j}$ ve $P_{i \times j}$, sırası ile $i \times j$ konumundaki pikselin gerçek değerini ve tahmin değerini temsil etmektedir. L değeri ise ortalama hata değerini göstermektedir.

4.1.2 Deneysel çalışmalar ve sonuçları

Deneysel çalışmalarda önerilen Inc-ZÖEA mimarisi LinkNet [64], PSPNet [134], FPNet [66] ve UNet [67] mimarileriyle kıyaslanmıştır.

Bu bölümde öncelikle önerilen modelin ve diğer ağların eğitimi ve test için parametre detayları verilmiştir. Daha sonra her veri kümesinin sayısal ve görsel sonuçları alt başlıklarda verilmiş ve değerlendirilmiştir.

4.1.2.1 Veri kümeleri ve uygulama detayları

Önerilen Inc-ZÖEA modelin performansını değerlendirmek için DAGM, MVTec-Doku ve MT veri kümeleri kullanılmıştır. Deneysel çalışmalarda kullanılan veri kümeleri ve eğitim/test için kullanılan örnek sayıları Çizelge 4.3'te verilmiştir.

Çizelge 4.3: Inc-ZÖEA modelin performansını değerlendirmek için kullanılan veri kümeleri ve örnek sayıları.

Veri kümesi	Eğitim Örnek Sayısı	Test Örnek Sayısı
	Hatalı/Hatasız/Toplam	Hatalı/Hatasız/Toplam
MT	289/289/578	97/97/194
DAGM	598/598/1196	602/602/1204
MVTec-Doku	285/1047/1332	97/352/449

Deneysel çalışmalarda otomatik öğrenme oranını ayarlamak için keras kütüphanesinde bulunan ReduceLROnPlateau modeli kullanılmıştır. ReduceLROnPlateau modeli kayıp (loss) değerini kontrol ederek öğrenme oranını ayarlamaktadır. Derin modelin eğitimi sırasında, belirli adım sayısı kadar (*patience*) kayıp değerinde bir değişme ya da düşüş olmadığında öğrenme oranı bir faktör (*factor*) değeri ile çarpılmaktadır. Bu sayede yeni öğrenme oranı ($lr_{new} = lr_{old} * factor$) belirlenmektedir.

Uygulamalarda model eğitim parametreleri aşağıdaki gibi ayarlanmıştır.

- Epoch:100
- Batch boyutu:16
- *factor*: 0,1
- *patience*: 8

- Başlangıç öğrenme oranı: 0,001
- Optimizasyonu: Adam optimizasyonu yöntemi.

4.1.2.2 Deneysel sonuçlar

MT, MVTEC-Doku ve DAGM veri kümesi kullanılarak elde edilen performans sonuçları Çizelge 4.4'te verilmiştir. Verilen sonuçlara göre 3 veri kümesinde, önerilen Inc-ZÖEA mimarisi diğer yöntemlere göre daha iyi sonuçlar sağlamıştır.

Çizelge 4.4: MT, MVTEC-Doku ve DAGM veri kümesi kullanılarak elde edilen performans sonuçları.

Veri Kümesi	MT		MVTEC-Doku		DAGM	
	mIoU	F1-skor	mIoU	F1-skor	mIoU	F1-skor
LinkNet	68,27	81,10	71,84	83,61	76,48	86,67
PSPNet	59,37	74,50	68,24	81,12	76,19	86,48
UNet	60,55	75,43	73,47	84,70	76,86	86,92
FPNet	65,87	79,42	65,70	79,30	75,43	85,99
Inc-ZÖEA	77,44	87,28	81,20	89,62	79,46	88,55

Çizelge 4.4'teki MT sonuçlarında, Inc-ZÖEA mimarisi %87,28 F1-skor ve %77,44 mIoU ile en yüksek başarıyı sağlarken ikinci en yüksek başarıyı LinkNet mimarisi sağlamıştır. ÖPA modelini kullanan Inc-ZÖEA ve FPNet modellerinin sonuçları arasında %11,57 mIoU ve 7,86 F1-skor farkı, Inc-ZÖEA'de bulunan ÇSÖÇ, MSKU vb. özel yapılarının katkılarını göstermektedir. Atlama bağlantıları kullanan LinkNet, UNet ve PSPNet mimariler arasında LinkNet mimarisi %68,27 mIoU skor ile yüksek başarı sağlarken PSPNet mimarisi %59,37 ile en düşük performans sonuçları üretmiştir. Çizelge 4.4'te verilen MVTEC veri kümesinin sonuçlarında MT veri kümesinde olduğu gibi Inc-ZÖEA mimarisi %81,2 mIoU ve %89,62 F1-skoru ile en yüksek başarıyı sağlamıştır. İkinci en yüksek performansı UNet ağı sağlamıştır. DAGM veri kümesinin sonuçlarında ise LinkNet, PSPNet, UNet ve FPNet ağları %75,43-%76,86 arasında yakın skorlar sağlamıştır.

	Görüntü	Gerçek Değer (GD)	LinkNet	PSPNet	Unet	FPNet	Inc-ZÖEA
Örn: 1		GD	iou:1.78	iou:0.0	iou:50.38	iou:0.0	iou:69.86
Örn: 2		GD	iou:16.27	iou:0.0	iou:52.47	iou:0.0	iou:61.16
Örn: 3		GD	iou:25.74	iou:56.29	iou:77.05	iou:65.11	iou:74.05
Örn: 4		GD	iou:67.74	iou:0.0	iou:73.68	iou:0.0	iou:78.78
Örn: 5		GD	iou:54.33	iou:26.53	iou:33.75	iou:32.23	iou:89.39
Örn: 6		GD	iou:74.09	iou:37.84	iou:71.78	iou:84.57	iou:92.31

Şekil 4.3: MT veri kümelerinden alınan örnek görsel sonuçlar.

Şekil 4.3'te MT veri kümesinden alınan bazı görüntüler ve tahmin sonuçları sunulmuştur. Şekil 4.3'te verilen 1 ve 2 numaralı örneklerde küçük boyutta hatalar bulunmaktadır. Bu örneklerde, Inc-ZÖEA mimarisi kullanılarak yüzey hataları en yüksek başarı ile tespit edilmiştir. Bunun yanı sıra 1 ve 2 nolu örneklerde, UNet modeli ile hatalar kısmen tespit edilirken diğer mimariler ile yüzey hatası tespit edilememiştir. MT'den alınan 3, 4, 5 ve 6 nolu örneklerde yüzey hataları arka plan ile aşırı benzerlik göstermektedir. Bu problem karşısında diğer mimarilere göre Inc-ZÖEA mimarisi daha başarılı sonuçlar vermiştir.

	Görüntü	Gerçek Değer (GD)	LinkNet	PSPNet	Unet	FPNet	Inc-ZÖEA
Örn: 1							
		GD	iou:29.04	iou:11.74	iou:49.51	iou:23.42	iou:77.28
Örn: 2							
		GD	iou:36.03	iou:5.0	iou:27.24	iou:12.94	iou:61.79
Örn: 3							
		GD	iou:24.96	iou:16.17	iou:33.81	iou:11.06	iou:41.65
Örn: 4							
		GD	iou:58.06	iou:8.92	iou:33.54	iou:34.78	iou:55.55
Örn: 5							
		GD	iou:0.0	iou:0.0	iou:12.56	iou:0.0	iou:45.45

Şekil 4.4: MVTEc-Doku veri kümelerinden alınan örnek görsel sonuçlar.

MVTEc veri kümesinin içerdiği hatalar genellikle küçük boyutlu ve karmaşık arka plana sahiptir. Şekil 4.4'te verilen MVTEc-Doku'nun 2, 3 ve 5 nolu örnekleri incelendiğinde dinamik (kompleks) arka plana sahip örneklerdir. Bu örneklerde Inc-ZÖEA mimarisi diğer mimarilere göre daha başarılı sonuçlar üretmiştir. Öte yandan örnek 4'te verilen yüzey dokusu homojen bir yapıya sahip ve nokta şeklinde bir yüzey hatası vardır. Bu örnekte tüm modeller yüzey hatasını bulmuştur. Modeller arasında yüzey hatası en iyi LinkNet ağı ile tespit edilmiştir. Örnek 1'deki yüzey hatası ise diğer örneklere göre belirgin olmasına rağmen arka plan ile benzer dokuya sahiptir. Bundan dolayı bu örnekte Unet ve Inc-ZÖEA modeli hatanın büyük bir bölümünü tespit ederken diğer modeller kısmen yüzey hatasını tespit etmiştir.

Sonuç olarak Şekil 4.3'te ve Şekil 4.4'te verilen görsel sonuçlarda önerilen Inc-ZÖEA mimarisi diğer modellere göre üstün olduğu gözlemlenmiştir.

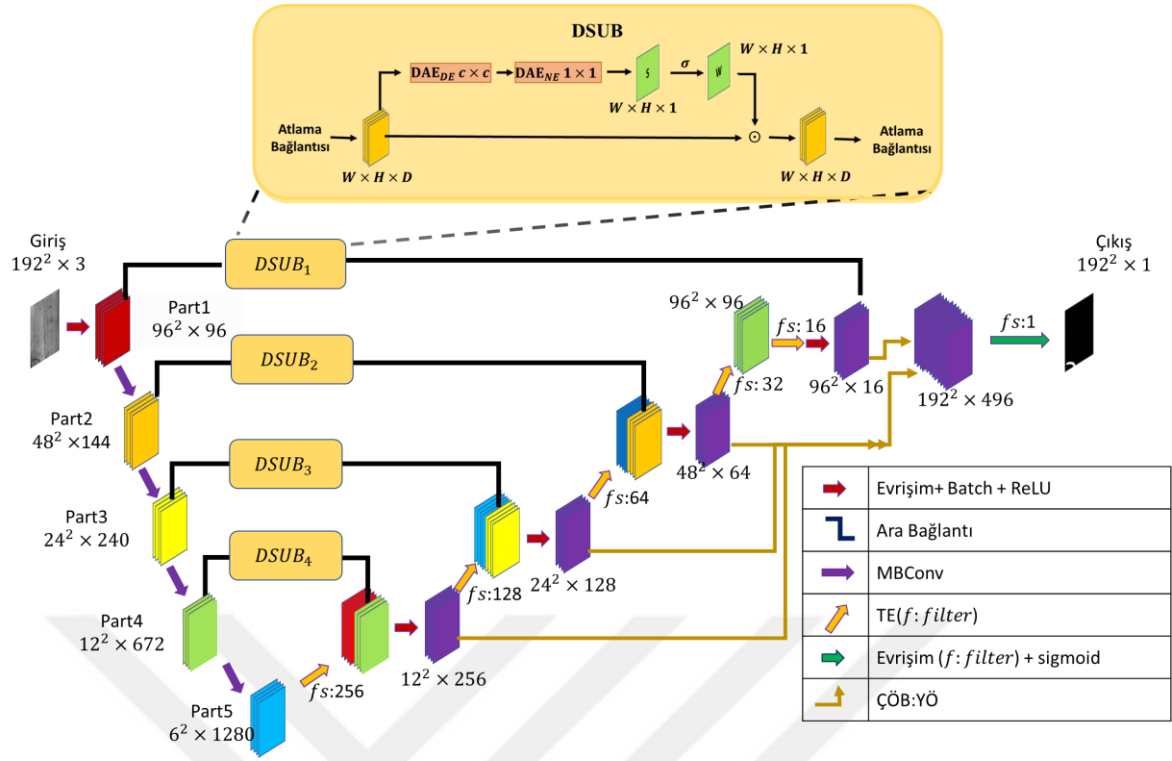
4.2 Uygulama 2: Derinlemesine Sıkma ve Uyarlama Bloğu Tabanlı Efficient-Unet Mimarisi

UNet ve FPNet gibi piksel tabanlı sınıflandırma için geliştirilen kodlayıcı-kod çözücü modellerde, genellikle atlama bağlantıları kullanılmıştır. Bu atlama bağlantıları sayesinde, kollayıcı bölümündeki düşük seviyeli öznitelikleri, kod çözücü bölümündeki yüksek seviyeli öznitelikler ile birleştirilerek zengin öznitelik haritaları elde edilmiştir. Kenar, renk ve doku gibi mekânsal bilgileri taşıyan düşük seviyeli öznitelik haritaları, hata tespitinde önemli olmasına rağmen, doğrudan kullanımları, yüksek seviyeli özniteliklerde bazı kritik bilgileri bastırabilmektedir. Bunun yanı sıra, elde edilen çok boyutlu öznitelik haritalarında küçük mekânsal ayrıntılar kaybolabilmektedir. Bu problemde yola çıkarak bazı çalışmalarda [10, 135, 136], atlama bağlantıları ile aktarılan öznitelik haritalarındaki önemli bilgileri ortaya çıkartmak için sıkma-uyarlama blokları kullanılmıştır.

Bu uygulamada güçlü bir kodlayıcı ve etkili bir atlama blok yapısı geliştirilmesi hedeflenmiştir. Bu hedef doğrultusunda Derinlemesine Sıkma ve Uyarlama Bloğu tabanlı Efficient-UNet (DSUB-EUNet) mimarisi geliştirilmiştir. Önerilen modelde kodlayıcı biriminde son zamanda oldukça popüler olan EfficientNet ağı kullanılmıştır. Bunun yanı sıra önerilen model için düşük seviyeli özniteliklerin ağırlıklandırılarak kullanılmasını sağlayan, yeni bir Sıkma-Uyarlama yapısı geliştirilmiştir. Derinlemesine Sıkma ve Uyarlama Bloğu (DSUB) olarak adlandırılan bu model, düşük bir işlem maliyeti harcayarak düşük seviyeli özniteliklerin daha verimli kullanılmasını sağlamıştır. Ayrıca önerilen DSUB-EUNet modelinde ara katmanlarda bulunan önemli öznitelikleri kaybetmeden taşımak için FPNet modelini temel alarak Çok Seviyeli Özellik Birleştirme (ÇÖB, Multi-level Feature Concatenated) yapısı önerilmiştir. ÇÖB yapısında, geleneksel UNet mimarisinden farklı olarak her seviyede elde edilen özniteliklerin son katmanda birleştirilmesine dayanmaktadır.

4.2.1 Önerilen yöntem

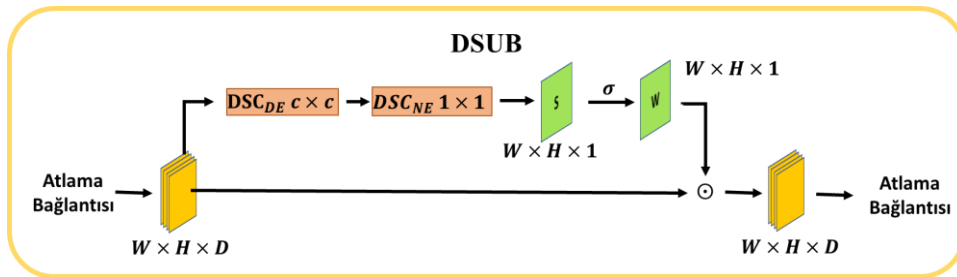
Önerilen DSUB-EUNet (Şekil 4.5) ağı üç bölüm altında detaylandırılmıştır: 1) Düşük seviyeli öznitelik değerlendirmesi için Derinlemesine Sıkma ve Uyarlama Bloğu, 2) Derinlemesine Sıkma ve Uyarlama Bloğu tabanlı Efficient UNet (DSUB-EUNet) mimarisi, 3) DSUB-EUNet Eğitim Prosedürü.



Şekil 4.5: DSUB-EUNet mimarisinin temel yapısı.

4.2.1.1 Derinlemesine sıkma ve uyarlama bloğu

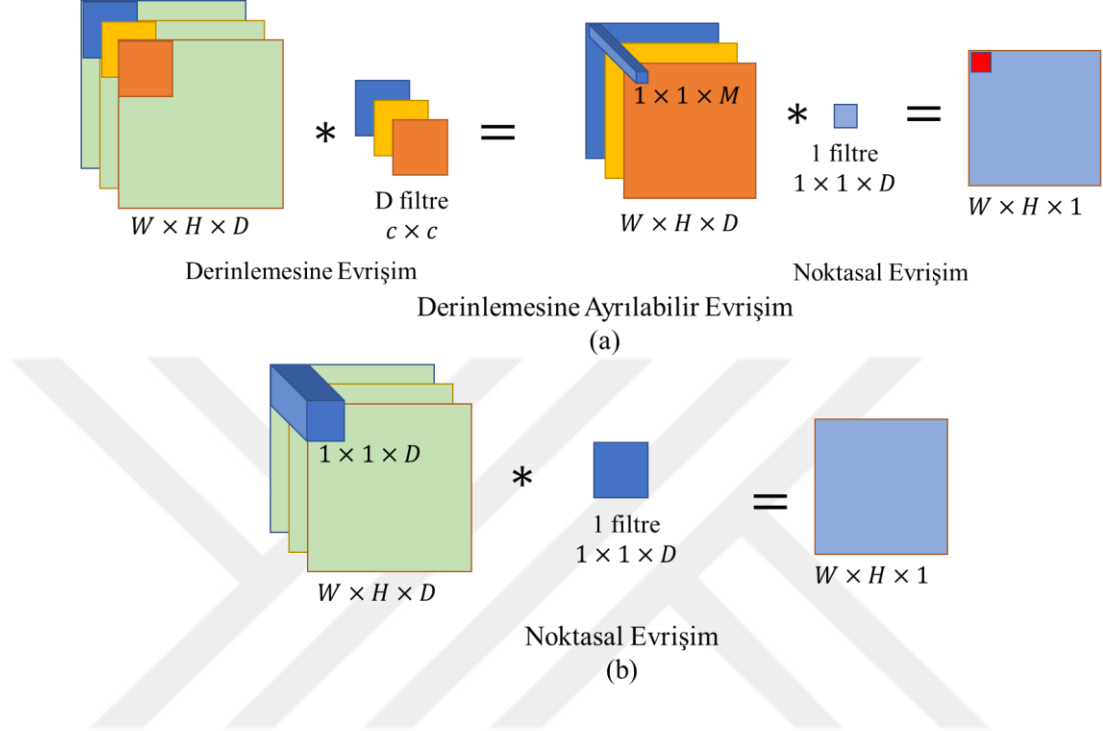
Sıkma ve Uyarlama (SU) blok yapıları öznetelik haritasında önemli ayrıntıları güçlendirmek ve ortaya çıkarmak için geliştirilmiş etkili yapılardır. Bu uygulamada, SU yapısını temel olarak Derinlemesine Sıkma ve Uyarlama Bloğu (DSUB) geliştirilmiştir. DSUB modelinde sıkma ağırlıkları oluşturmak için çekirdek boyutu 3×3 olan, derinlemesine ayrılabilir bir evrişim yapısı kullanılmıştır. Şekil 4.6'da önerilen DSUB yapısı gösterilmiştir.



Şekil 4.6: Önerilen DSUB yaklaşımının yapısı (NE:Noktasal Evrişim, DE: Derinlemesine Evrişim).

Mekânsal bilgileri ön plana taşımak için geliştirilen Kanal Sıkma ve Mekânsal Uyarlama (KSMU) [105, 107] modelinde, (detayları Bölüm 2.3.1'de verilmiştir.) çekirdek boyutu 1×1 olan noktasal evrişim katmanı kullanılmıştır. Fakat evrişim işlemi aslında bir filtreleme işlemi olduğu için evrişim katmanlarından sonra elde edilen öznetelikler genellikle

çevresindeki öznitelikler ile yakın bilgiler içermektedir. Mekânsal bilgilerin ağırlıklarını çıkartmak için KSMU’da olduğu gibi doğrudan noktasal evrişim kullanılması özniteliklerin sadece derinlik boyutu ile ilişkilendirilebilmektedir. Bu problemden yola çıkarak önerilen DSUB modelinde derinlemesine ayrılabilir evrişim modeli (Şekil 4.7) kullanılmıştır.



Şekil 4.7: DSUB modelinde kullanılan evrişim (a) ve Orijinal KSMU modelinde kullanılan (b) evrişim katman modelleri.

Şekil 4.6’da önerilen DSUB iki bölümden oluşmaktadır. İlk kısım, katman sıkma ağırlıklarını elde etmek için Derinlemesine Ayrılabilir Evrişim (DAE) yapısı kullanılmıştır. Şekil 4.7.a’de gösterildiği gibi bu modelde öncelikle $W \times H \times D$ boyutunda olan girdiye bir derinlemesine evrişim katmanı arkasından noktasal evrişim katmanı uygulanmıştır. İşlem sonucunda $W \times H \times 1$ boyutunda olan S ağırlık matrisi elde edilmiştir. Daha sonra S ağırlık matrisinin değerleri 0 ile 1 arasında sıkıştırmak için sigmoid fonksiyonu ($\sigma(\cdot)$) uygulanır. Bu işlem adımları Denklem 4.4’teki gibi formülize edilmiştir.

$$output_{DE(W \times H \times D)} = DAE_{DE(c \times c)}^{f_s=D} * F_{W \times H \times D}$$

$$S_{(W \times H \times 1)} = DAE_{NE(1 \times 1)}^{filters=1} * output_{DE(W \times H \times D)} + w_0 \quad (4.4)$$

$$UA_{(W \times H \times 1)} = \sigma(S) = \frac{1}{1 + e^{-S}}$$

Burada DAE_{NE} ve DAE_{DE} sırasıyla nokta bazında ve derinlik bazında evrişimdir. c ve w_0 sırasıyla çekirdek boyutu ve bias değerini gösterir. $W \times H \times D$ giriş öznitelik haritasının boyutudur. $F_{W \times H \times D}$ ise DAE girişine verilen öznitelik haritasıdır. Son olarak UA uyarlama için elde edilen ağırlıklardır.

DSUB'nin ikinci bölümünde, UA matrisinin değerleri Keras'ta bulunan lambda katmanı ile ardışık olarak tekrarlanmakta ve giriş öznitelik boyutuna getirilmektedir. İşlem sonunda F giriş öznitelik haritası ile aynı boyutta olan ve değerleri 0 ile 1 arasında değişen genişletilmiş uyarlama ağırlıkları (GUA) elde edilmektedir. DSUB'nin son bölümünde, GUA ile girdi değerleri arasında eleman bazında çarpma işlemi gerçekleştirilmektedir (Denklem 4.5).

$$F_{new} = F \otimes GUA \quad (4.5)$$

Burada \otimes eleman bazında çarpmayı göstermektedir. Sonuç olarak, DSUB çıktısında F_{new} ile temsil edilen ve F giriş öznitelik haritasının ağırlıklandırılmış öznitelik haritası elde edilmektedir.

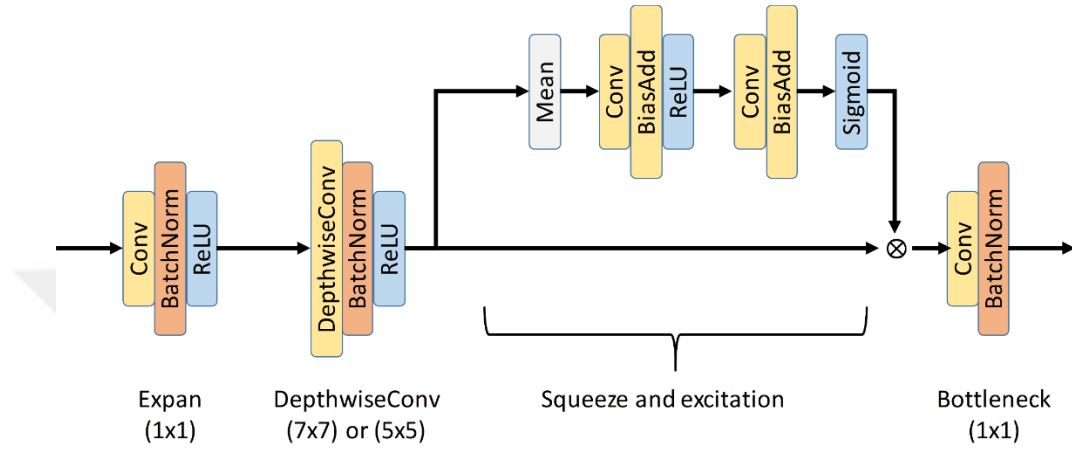
4.2.1.2 Derinlemesine sıkma ve uyarlama blok tabanlı Efficient UNet

Derinlemesine Sıkma ve Uyarlama Blok Tabanlı Efficient UNet (DSUB-EUNet) mimarisi temel olarak Unet modeli gibi kodlayıcı, kod çözücü ve atlama bağlantılarından oluşmaktadır. Bunun yanı sıra atlama bağlantılarına DSUB blokları eklenmiştir. Ayrıca önerilen modelin kod çözücü bölümündeki farklı seviyeli öznitelikleri doğrudan son katmanlara aktarmak için FPNet [66] modelinden esinlenerek Çok Seviyeli Öznitelik Birleştirme (ÇÖB, Multi-level Feature Concatenated Block) yapısı geliştirilmiştir.

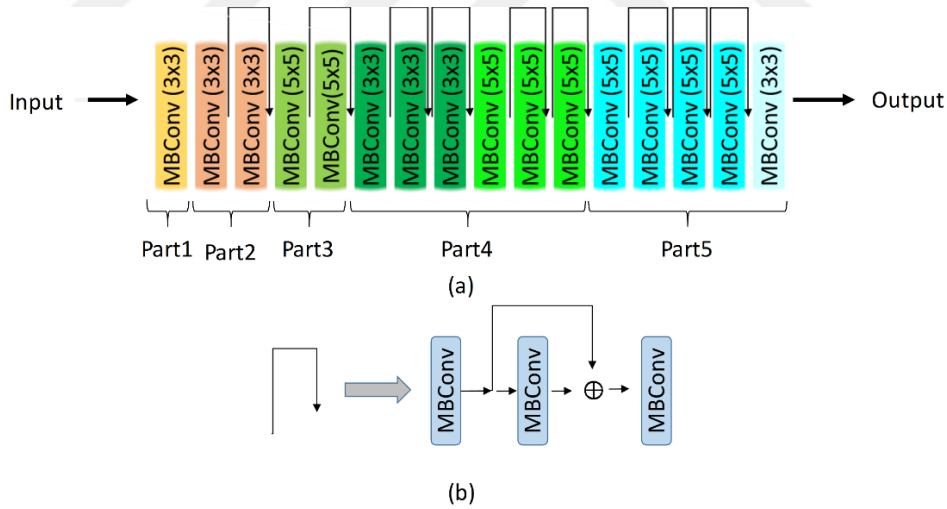
DSUB-EUNet mimarisinin kodlayıcı kısmında önceden eğitilmiş bir EfficientNetB0 ağı kullanılmıştır. Tan ve Le tarafından yapılan çalışmada [82], katman sayısı (derinlik), filtre sayısı (genişlik) ve görüntü çözünürlüğü gibi parametreleri doğru bir şekilde ayarlamak için mimari ölçekleme yaklaşımı önerilmiştir. Bu yaklaşımda genişlik, derinlik ve çözünürlük parametrelerini dengeli bir şekilde ölçeklendirerek optimum hiper parametre değerlerine ulaşılması hedeflenmiştir. Bu bakış açısıyla EfficientNet ağları geliştirilmiştir. EfficientNet ağları ImageNet veri kümesinde oldukça yüksek başarılar göstermiştir [82].

EfficientNet ağları, MBConv [85, 95, 137] (Şekil 4.8) blok yapılarından oluşturulmuştur. Tan ve Le yaptıkları çalışmada [82], Farklı sayıda MBConv katmanı içeren sekiz farklı EfficientNet (B0-B7) ağı geliştirilmiştir. DSUB-EUNet ağının kodlayıcı kısmında en hızlı

ve en az parametre içeren EfficientNetB0 ağı modeli kullanılmıştır. EfficientNetB0 ağı, 16 adet MBConv katmanı içermektedir. Bu katmanlar arasındaki bağlantılar Şekil 4.9'da gösterilmiştir. Önerilen DSUB-EUNet ağının atlama bağlantısı için EfficientNetB0 ağı dört bölüme (Şekil 4.9.a) ayrılmıştır. Her parçanın çıkışı atlamalı bağlantılar ile kod çözücü bölümüne aktarılmıştır. Şekil 4.5'te gösterildiği gibi DSUB-EUNet'in her atlama bağlantısına bir DSUB eklenmiştir.

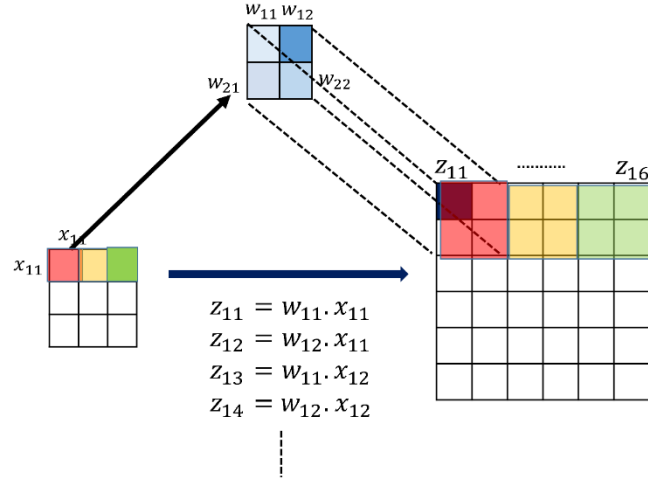


Şekil 4.8: MBConv Bloğunun Yapısı



Şekil 4.9: (a) EfficientNetB0 ağının ayrıntıları, (b) artık bağlantı.

DSUB-EUNet'in kod çözücü kısmı orijinal UNet'teki gibi 4 seviyeden oluşmaktadır. Bunun yanı sıra UNet'ten farklı olarak boyut yükseltme katmanı için Ters Evrişim (TE, Conv2DTranspose) katmanı kullanılmıştır. TE katmanı, ters evrişim sürecini uygulayarak girdi verilerinin boyutunu yükseltmektedir. Eğitim sırasında veri boyutunu arttırmak için en uygun filtre ağırlıklarını öğrenmektedir. Ters Evrişim yönteminin ayrıntıları Şekil 4.10'da gösterilmiştir.



Şekil 4.10: Ters Evrişim katmanı; x katmanının girdi verileri, z katmanının çıktıları ve w filtrenin ağırlık değerleridir.

Kodlayıcının her seviyesindeki öznitelikleri doğrudan kullanmak için önerilen ÇÖB'de ilk olarak, öznitelik haritaları yukarı örnekleme katmanı ile aynı boyuta yükseltilmiştir. Daha sonra bu öznitelikler kod çözücü bölümünün son katmanında birleştirilmiştir (Denklem 4.6).

$$M = \text{birleştirme} \begin{pmatrix} F_{DP\ 1}^{96^2 \times 16} \\ Y\ddot{O}^{ag=2} \left(F_{DP\ 2}^{48^2 \times 64} \right) \\ Y\ddot{O}^{ag=4} \left(F_{DP\ 3}^{24^2 \times 128} \right) \\ Y\ddot{O}^{ag=8} \left(F_{DP\ 4}^{12^2 \times 256} \right) \end{pmatrix} \quad (4.6)$$

Burada, M birleştirilmiş öznitelik haritasıdır; $Y\ddot{O}^{ad}$ adım değeri ad olan yukarı örnekleme katmanıdır. $F_{DP\ i}^S$ ise DSUB-EUNet'in kod çözücü bölümündeki i 'nci seviyesindeki S boyutlu öznitelik haritasıdır. DSUB-EUNet'in son kısmında elde edilen M öznitelik haritası bir evrişim ve sigmoid fonksiyonu uygulanarak yüzey hata tespit tahmin haritası elde edilmiştir.

4.2.1.3 Önerilen DSUB-EUNet mimarisinin eğitim prosedürü

Derin öğrenme tabanlı çalışmalarda büyük veri kümelerinin kullanılması, model performansı için çok önemlidir. Küçük veya tek tip veriler kullanılarak eğitilen modeller, genellikle aşırı öğrenme veya modelin eğitilememesi gibi problemler oluşturmaktadır. Bu problemler modelin düşük performans göstermesine neden olmaktadır. Bu sorunla başa çıkmak için yaygın olarak kullanılan yöntemler, önceden eğitilmiş ağların kullanılması (Transfer öğrenme (TÖ)) ve Veri Arttırma (VA) yöntemleri ile veri kümelerindeki örnek sayısının artırılmasıdır [86, 138]. VA yöntemleri, çevirme, yansıtma ve kırpma işlemleri

yoluyla örnek sayısını arttırmak için kullanılmaktadır. Bunun yanı sıra bulanıklaştırma veya görüntü bozma gibi yaklaşımlarda vardır. VA yöntemleri veri kümesindeki örnek sayılarını arttırsa da sınıflar arası [86] benzerliklerin çok olduğu veri kümelerinde bu yöntemler yetersiz kalabilmektedir. Ayrıca bulanıklaştırma ve görüntü bozma gibi VA yöntemleri, küçük boyutlu yüzey hatalarının ve arka plana benzer olan örneklerde önemli ayrıntılarının kaybolmasına neden olabilmektedir. Bu nedenle transfer öğrenme yaklaşımı küçük veri kümelerinde VA yönteminden daha iyi sonuçlar elde edebilmektedir [7, 85]. Transfer öğrenme yaklaşımı Bölüm 2.4'te detaylandırıldığı gibi, bir problemi çözmek için geliştirilen bir modelin ağırlıklarını, rastgele başlatmak yerine önceden eğitilmiş bir modelin ağırlıkları ile başlatmaktır. Transfer öğrenme birçok çalışmada yaygın olarak kullanılmaktadır [7, 13, 86, 138].

Bu çalışmada, önerilen DSUB-EUNet mimarisinin enkoder bölümünde başlangıç ağırlıkları, ImageNet veri kümesi ile eğitilmiş EfficientNetB0 mimarisinin ağırlıkları kullanılmıştır. Geri kalan ağırlıklar ise rastgele değerler atanmıştır. Eğitim boyunca parametreler en iyi sonucu vermek için güncellenmiştir. Önerilen DSUB-EUNet ağının eğitiminde İkili Çapraz Entropi (Binary Cross-Entropy, Denklem 4.3) kayıp fonksiyonu kullanılmıştır.

4.2.2 Deneysel çalışmalar ve sonuçları

Deneysel çalışmalarda MT [69], AITEX [133] ve DAGM [139] veri kümeleri kullanılmıştır. Ayrıca, önerilen ağ mimarisi LinkNet [64], PSPNet [134], FPNNet [66], U-net [67], VGG-UNet [56], Mobil-UNet [86], Eff-UNet [85], Attention-UNet [104], Trans-UNet [116], ve Swin-UNet [118] gibi son teknoloji yaklaşımlar ile kıyaslanmıştır.

Bu bölümde öncelikle derin mimarilerinin eğitimi ve testi için parametre detayları verilmiştir. Daha sonra her veri kümesinin sayısal ve görsel sonuçları alt başlıklarda verilmiş ve değerlendirilmiştir.

4.2.2.1 Veri kümeleri ve uygulama detayları

Önerilen model Python platformunda TensorFlow ve Keras kütüphaneleri kullanarak geliştirmiştir. Önerilen modelin eğitim aşamasında doğru öğrenme oranını ve yeterli epoch sayılarını belirlemek için Keras kütüphanesinde bulunan ReduceLROnPlateau ve EarlyStopping modelleri kullanılmıştır [123]. ReduceLROnPlateau modeli kayıp (loss) değerini kontrol ederek öğrenme oranını ayarlar. Önerilen modelin eğitimi sırasında, belirli adım sayısı kadar (*patience*) kayıp değerinde bir değişme ya da düşüş olmadığında

ReduceLROnPlateau modeli öğrenme oranını bir faktör (*factor*) değeri ile çarpmaktadır. Bu sayede yeni öğrenme oranı ($lr_{new} = lr_{old} * factor$) belirlenmektedir. EarlyStopping modeli ise eğitim sırasında, belirli adım sayısı kadar (*patience*) kayıp değerinde bir değişme ya da düşüş olmadığında eğitimi sonlandırmaktadır. Bu çalışmada ReduceLROnPlateau fonksiyonunun *factor* ve *patience* değerleri sırası ile 0,1 ve 3 olarak ayarlanmıştır. EarlyStopping'ın *patience* değeri ise 8 olarak alınmıştır. Ayrıca eğitim aşamasında Batch boyutu 16 ve başlangıç öğrenme oranı 0,001 alınmıştır.

Önerilen DESB yapısının *c* değeri için 3, 5 ve 7 değerleri kullanılmıştır. Deneysel çalışmalarda piksel seviyesinde yüzey hata tespiti için en iyi performans sonuçları *c*=3 değeri ile bulunmuştur.

Deneysel çalışmalar, MT ve AITEX veri kümelerinin %50'si eğitim için ve %50'si test için ayrılmıştır (Çizelge 4.5). DAGM veri kümesinde eğitim ve test görüntüleri zaten ayrı olduğundan orijinal olarak kullanılmıştır. Modellerin daha yüksek başarı oranları elde etmesi için eğitim veri kümesindeki her bir örnek döndürme, yansıtma ve kırpma gibi veri arttırma yöntemleri uygulanarak eğitim veri kümesindeki örnek sayısı sekiz kat arttırılmıştır.

Çizelge 4.5: MT, DAGM ve AITEX veri kümelerinin eğitim ve test örnekleri.

Veri kümesi	Eğitim verileri			Test verisi		
	Hatasız	Hatalı	Toplam	Hatasız	Hatalı	Toplam
MT	479	193	672	479	193	672
DAGM	3004	446	3450	2996	454	3450
AITEX	163	82	245	163	82	245

4.2.2.2 MT veri kümesinin sonuçları

MT veri kümesinde, küçük boyutlu ve arka plana benzer yüzey hatalarını içeren görüntüler bulunmaktadır. Bunun yanı sıra değişen ışık koşullarında çekilen görüntüleri içermektedir. Bu nedenle, yüzey hata tespiti için zor bir veri kümesidir. MT veri kümesindeki bazı yüzey örneklerinin ve yöntemlerinin sayısal (IoU) ve görsel tahminleri Şekil 4.11'de verilmiştir. Şekil 4.11'deki 1, 2 ve 3 nolu örneklerde küçük yüzey hataları içermektedir. Mobile-UNet ve Eff-UNet mimarileri güçlü katman yapılarına sahip olmasına rağmen, bu tür örnekler için zayıf sonuçlar üretmiştir. Bu durumun temel nedeni, bu ağ mimarilerinin birçok katman içermesi ve bu nedenle bu tür hataları ayırt etmeye yardımcı olabilecek küçük ayrıntıları kaybetmesidir. Ancak, önceden eğitilmiş Efficient ağına dayalı önerilen DSUB-EUNet mimarisi, içerdiği DSUB sayesinde elde edilen önemli öznelikleri

korumuş ve bu öznetelikleri ÇÖB yapısı aracılığıyla son katmana taşımıştır. Böylece önerilen model bu tür örneklerde yüksek performans elde etmiştir. Öte yandan UNet'teki atlama bağlantılarındaki öznetelikleri güçlendirmek için geliştirilen Attention-UNet modeli 2. sıradaki örnekte DSUB-EUNet modeline yakın sonuçlar elde ederken, 1.ve 3. Sıradaki örneklerde düşük sonuçlar vermiştir. Bunun yanı sıra 1., 2. ve 3. sıradaki örneklerde küçük hataları UNet ve VGG-UNet mimarileri Mobile-UNet ve Eff-UNet mimarilerinden daha iyi tespit etmiştir.

Görüntü	Gerçek Değer	LinkNet	PSPNet	FPNet	U-Net	VGG-UNet	Mobile-UNet	Trans-UNet	SwIn-UNet	Attention-UNet	Eff-UNet	DSUB-EUNet
1		iou:0.0	iou:0.0	iou:0.0	iou:48.27	iou:72.0	iou:88.46	iou:51.11	iou:0.0	iou:31.11	iou:0.0	iou:86.0
2		iou:0.0	iou:8.02	iou:0.0	iou:44.03	iou:17.85	iou:14.94	iou:86.84	iou:0.0	iou:75.28	iou:13.15	iou:75.53
3		iou:40.49	iou:0.0	iou:0.0	iou:56.06	iou:87.25	iou:32.89	iou:66.93	iou:5.65	iou:54.23	iou:64.22	iou:73.8
4		iou:0.0	iou:81.85	iou:82.44	iou:0.0	iou:0.0	iou:34.58	iou:0.0	iou:0.0	iou:0.0	iou:0.02	iou:89.68
5		iou:0.0	iou:1.86	iou:18.48	iou:0.0	iou:0.0	iou:78.52	iou:36.74	iou:0.0	iou:62.82	iou:37.76	iou:83.74
6		iou:0.0	iou:0.0	iou:0.0	iou:49.26	iou:29.5	iou:44.66	iou:49.56	iou:17.54	iou:61.88	iou:52.25	iou:60.73
7		iou:12.04	iou:0.0	iou:0.0	iou:81.66	iou:7.18	iou:83.17	iou:87.64	iou:0.86	iou:85.99	iou:80.48	iou:85.57
8		iou:2.26	iou:82.63	iou:78.97	iou:49.42	iou:58.21	iou:74.58	iou:75.11	iou:0.08	iou:58.88	iou:64.84	iou:83.42

Şekil 4.11: MT veri kümesindeki görsel sonuçları.

MT veri kümesinin ikinci ana zorluğu, Şekil 4.11'deki 4., 5., 6. ve 7. sıradaki örneklerde görüldüğü gibi, arka plana son derece benzer hata tiplerine sahip olmasıdır. Bu tür hataların tespiti, arka plan ve yüzey hatasını ayırabilecek güçlü öznetelikler gerektirmektedir. Önerilen DSUB-EUNet mimarisi, içerdiği Efficient ağı, DSUB ve ÇÖB yapıları sayesinde güçlü öznetelikler elde etmiş ve bu özneteliklerdeki önemli detayları korumuştur. Böylece bu tür hataları içeren görüntülerde en yüksek başarıyı elde etmiştir. Ayrıca 5., 6. ve 7. sıradaki örneklerde Attention-UNet mimarisi Mobile-UNet ve Eff-UNet'e göre daha yüksek başarı elde etmiştir. 7. sırada Trans-UNet ağı en iyi skorla hatayı tespit ederken, 5. ve 6. sırada Attention-UNet ve önerilen DSUB-EUNet modelinden daha düşük sonuçlar elde etmiştir.

Swin-UNet ağı, bu örneklerde yüzey hatalarını tespit edememiştir. Bunun temel sebebi dönüştürücülerin eğitimi için güçlü bir veri kümesine ihtiyaç duyması ve dönüştürücülerinin küçük mekânsal ayrıntıları yakalamadığından kaynaklanmaktadır [116]. Bu bakış açısıyla geliştirilen Trans-UNet ağında, ilk katmanlardaki mekânsal detayları korumak için evrişim katmanını kullanılırken UNet'in alt katmanlarında dönüştürücüler kullanılmıştır. Bu yapı sayesinde Trans-UNet ağı 2., 3., 7. ve 8. sıralarda yüksek puanlar elde etmiştir.

Sonuç olarak, önerilen DSUB-UNet mimarisi güçlü yapısı sayesinde Şekil 4.11'deki örneklerde en yüksek performansları göstermiştir. Önerilen modelin ardından en iyi sonuçlar Mobile-UNet ve Trans-UNet mimarileri ile elde edilmiştir.

Çizelge 4.6: MT veri kümesinden elde edilen test sonuçları.

Model	mIoU	F1 Skor	AUC
LinkNet	16,04	27,65	92,09
PSPNet	11,32	20,34	70,67
FPNet	52,99	82,08	96,84
U ağı	51,98	46,93	93,62
VGG-UNet	54,68	70,70	97,76
Mobil-UNet	76,29	86,55	99,31
Trans-UNet	64,69	78,56	94,96
Swin-UNet	11,86	21,21	72,14
Attention-UNet	61,18	75,92	97,51
Eff-UNet	65,08	78,85	98,82
DSUB-EUNet	80,51	89,20	99,51

Çizelge 4.6'da, MT veri kümesi ile gerçekleştirilen deneysel çalışmalardan elde edilen performans sonuçları verilmiştir. Önerilen DSUB-EUNet %89,20 F1 skoru ve %80,51 mIoU ile en yüksek başarı oranına ulaşmıştır. Mobile-UNet, %86,55 F1 skoru ve %76,29 mIoU ile ikinci en yüksek başarı oranını elde etmiştir. Eff-UNet mimarisi önceden eğitilmiş EfficientNet mimarisini kullanmasına rağmen önemli öznetelikleri koruyamadığı için Eff-UNet mimarisi %65,08 mIoU ile üçüncü sırada yer almıştır. Buna kıyasla Eff-UNet mimarisindeki gibi EfficientNet modelini temel alınarak geliştirilen DSUB-EUNet, içerdiği DSUB ve ÇÖB modelleri sayesinde daha yüksek skorlara ulaşmıştır. Çizelge 4.6'da gösterilen yöntemlerden en düşük mIoU sonuçları LinkNet (%16,04), Swin-UNet (%11,86) ve PSPNet (%11,32) ağları ile elde edilmiştir.

4.2.2.3 DAGM veri kümesi sonuçları

DAGM veri kümesi, yüzey hata tespiti için geliştirilmiş sentetik bir veri kümesidir. DAGM veri kümesinden alınan bazı örnek görüntüler ve yöntemlerin tahmin sonuçları Şekil

4.12’de verilmiştir. Şekil 4.12’de verilen örneklerde, DSUB-EUNet mimarisi ile daha yüksek bir başarı elde edilmiştir. Şekil 4.12’deki 3., 4. ve 8. sıradaki örneklerin arka planı düzenli olduğu için, PSPNet, FPNNet ve UNet dışındaki diğer yöntemler başarılı olmuştur. Öte yandan 1., 2., 5., 6. ve 7. sıradaki örneklerde Trans-UNet, Attention-UNet ve önerilen DSUB-EUNet dışındaki diğer tüm yöntemler düşük sonuçlar üretmiştir.

Image	Gerçek Değer	LinkNet	PSPNet	FPNet	U-Net	VGG-Unet	Mobile-Unet	Trans-Unet	Swin-Unet	Attention-Unet	Eff-Unet	DSUB-EUNet
1												
2												
3												
4												
5												
6												
7												
8												

Şekil 4.12: DAGM veri kümesindeki görsel sonuçlar.





























































DeneySEL çalışmalarda elde edilen DAGM veri kümesinin sayısal sonuçları Çizelge 4.7’de verilmiştir. Bu sonuçlarda DSUB-EUNet mimarisi kullanılarak en yüksek performans (%75,40 mIoU ve %85,97 F1 skoru) sonuçları elde edilmiştir. EFF-UNet mimarisi, %71,80 mIoU ve %83,58 F1 skoru ile ikinci en yüksek başarıyı sağlamıştır. Önceden eğitilmiş EfficientNet mimarisinin parametrelerini kullanan DSUB-EUNet ve Eff-UNet mimarilerinin karşılaştırıldığında, önerilen DSUB-EUNet mimarisi %3,6 mIoU ve %2,39 F1 skoru daha yüksek puanlar üretmiştir.

Çizelge 4.7: DAGM veri kümesinden elde edilen test sonuçları.

Model	mIoU	F1 Skoru	AUC
LinkNet	69,23	81,81	99,62
PSPNet	35,22	52,09	86,40
FPNet	35,30	52,18	85,76
U-Net	37,16	54,18	87,85
VGG-UNet	65,68	79,29	98,95
Mobil-UNet	69,64	82,10	99,59
Trans-UNet	63,30	77,52	97,03
Swin-UNet	39,57	56,70	86,10
Attention-UNet	69,58	82,06	99,01
Eff-UNet	71,80	83,58	99,79
DSUB-EUNet	75,40	85,97	99,77

4.2.2.4 AITEX veri kümesi sonuçları

AITEX veri kümesi arka plana oldukça benzer ve küçük boyutlu hatalar içermektedir. AITEX veri kümesinin bazı örnek görüntüleri ve yöntemlerin tahmin sonuçları Şekil 4.13'te verilmiştir. Şekil 4.13'te, önerilen DSUB-EUNet mimarisinin diğer yöntemlerden ortalama olarak daha başarılı sonuçlar verdiği görülmektedir. Şekil 4.13'teki 1., 2., ve 3. örnekte arka plandan ayırt edilmesi zor olan hataları içermektedir. Swin-UNet modeli ile 1. ve 2. örnekteki hataları tespit edilemezken diğer modeller ile kumaş hatası, yüksek performansla tespit edilmiştir. 3. sıradaki örnekte ise Attention-UNet, Eff-UNet ve DSUB-EUNet modelleri kullanılarak arka plana benzer hatalar yüksek doğrulukla tespit edilmişken diğer modeller düşük sonuçlar üretmiştir. 4. ve 5. sıradaki görüntülerde küçük boyutlu hatalar içermektedir. Bu görüntülerde, DSUB-EUNet, Mobile-UNet, Trans-UNet ve Eff-UNet modelleri diğer modellere göre daha başarılı sonuçlar üretmiştir.

Image	Gerçek Değer	LinkNet	PSPNet	FPNet	U-Net	VGG-UNet	Mobil-UNet	Trans-UNet	Swin-UNet	Attention-UNet	Eff-UNet	DSUB-EUNet
1												
2												
3												
4												
5												

Şekil 4.13: AITEX veri kümesindeki görsel sonuçlar.

AITEX veri kümesi için performans sonuçları Çizelge 4.8’de verilmiştir. MT ve DAGM veri kümelerinde olduğu gibi, AITEX veri kümesinde de en yüksek başarı oranı (%82,47 mIoU ve %90,39 F1 skoru) DSUB-UNet mimarisi kullanılarak elde edilmiştir. En yüksek ikinci performansı (%81,53 mIoU) ise Mobile-UNet ağı ile elde edilmiştir. Önceden eğitilmiş bir ağ modeli kullanmayan LinkNet, PSPNet ve U-net ağları ile %71,77 ve %80,20 arasında mIoU performansına ulaşılmıştır. Öte yandan Trans-UNet ve Attention-Unet modelleri kullanılarak sırasıyla %80,72 ve %80,32 mIoU başarıları skorları elde edilmiştir. Ayrıca Swin-UNet ağı ile AITEX veri kümesinde en düşük performans elde edilmiştir.

Çizelge 4.8: AITEX veri kümesinden elde edilen test sonuçları.

Model	mIoU	F1 skoru	AUC
LinkNet	71,77	83,57	99,45
PSPNet	77,43	87,28	98,54
FPNet	66,05	79,55	98,76
U ağı	80,20	89,01	99,78
VGG-UNet	79,63	88,66	99,76
Mobil-UNet	81,53	89,82	99,81
Trans-UNet	80,72	89,33	99,80
Swin-UNet	41,09	58,25	96,13
Attention-UNet	80,32	89,08	99,69
Eff-UNet	81,16	89,60	99,81
DSUB-EUNet	82,47	90,39	99,77

Sonuç olarak, deneysel çalışmalarda, önerilen DSUB-EUNet ağı, değişen hata boyutu, arka plana benzerliği ve karmaşık arka plan gibi yüzey hata problemleri ile başa çıkarak yüksek performanslar göstermiştir.

4.2.3 Tartışma ve analiz

Bu bölümde önerilen DSUB-EUNet mimarisinin bileşenlerini analiz etmek için birtakım deneysel çalışmalar yapılmıştır. Ayrıca önerilen model literatürdeki en son yöntemler ile karşılaştırılarak güçlü ve zayıf yönleri ortaya konulmuştur.

4.2.3.1 Önerilen modelin önceki çalışmalarla karşılaştırılması

Bu bölümde önerilen DSUB-EUNet mimarisi önceki çalışmalar ile kıyaslanmıştır. Literatürde bulunan ve MT, AITEX ve DAGM veri kümeleri kullanarak geliştirilen modellerin ve önerilen DSUB-EUNet modelinin performans sonuçları Çizelge 4.9’da verilmiştir. Çizelge 4.9’dan görüldüğü gibi, Önceden eğitilmiş VGG16 ağına dayanan PGA-Net[7] mimarisi kullanılarak, MT veri kümesinde %71,31 mIoU ve DAGM (1., 3., 5., 6., 7. ve 9. sınıf) veri kümesinde %74,78 mIoU elde edilmiştir. Buna kıyasla önerilen DSUB-

EUNet modeli kullanılarak MT veri kümesinde %80,51 mIoU ve DAGM (1., 3., 5., 6., 7. ve 9. sınıf) veri kümesinde %75,83 mIoU skorlarına ulaşılmıştır. Önceden eğitilmiş Resnet50 ağını kullanan ve PGA-Net'e benzer bir yapıda geliştiren DFF [41] yöntemi ise MT veri kümesinde %73,70 mIoU ve %82,20 F1-skoru sağlamıştır. Öte yandan MT veri kümesinin paylaşıldığı çalışmada geliştirilen MCuePushU [69] modeli ile %54,9 $F_{\beta=0,3}$ (F beta puanı) elde edilmiştir. Öte yandan DAGM veri kümesinin 1., 2., 3., 4., 5. ve 6. sınıflarını üzerinden geliştirilen HEF [55] ve SA-CAM [71] ağları sırası ile %73,03 mIoU ve %70,56 mIoU performansları sağlamıştır.

Çizelge 4.9: Yüzey hata tespiti için geliştirilen yöntemler ve sonuçları.

Veri kümesi	Yöntemler	mIoU	F1 skoru	$F_{\beta=0,3}$	Duyarlılık	Özgüllük
MT	PGA-Net[7]	71,31	-	-	-	-
	DFF[41]	73,70	82,20	80,10	-	-
	MCuePushU	-	-	54,90	-	-
	DPAE[140]	-	78,80	-	-	-
	DSUB-EUNet	80,51	89,20	92,19	99,56	92,80
DAGM (1,2,3,4,5 ve 6 sınıf)	HEF[55]	73,03	-	-	-	-
	SA-CAM [71]	70,56	74,46	-	-	-
	DSUB-EUNet	75,40	85,97	87,63	99,93	87,63
DAGM (1,3,5,6,7 ve 9 sınıfı)	PGA-Net[7]	74,78	-	-	-	-
	DSUB-EUNet	75,86	86,25	92,42	99,68	93,75
AITEEX	AITEX [133]	-	-	-	86,67	88,57
	VGS[141]	-	81,00	-	-	-
	DCSNet [142],	34,2	-	-	84,40	-
	CU-Net[108]	-	-	-	92,7	-
	IFV[143]	-	82,58	-	82,59	82,68
	DSUB-EUNet	82,47	90,39	90,73	99,84	90,80

AITEEX veri kümesinin geliştirildiği çalışmada [133], kumaş hata tespiti için Gabor filtreleri kullanılmıştır. Bu çalışma, bu veri kümesi için özel tasarlanmış Gabor yönteminin parametrelerini manuel olarak ayarlamıştır. Çizelge 4.9'daki karşılaştırma, önerilen DSUB-EUNet ağının Gabor yöntemine göre duyarlılık (sensitivity) ve özgüllük (specificity) performans sonuçlarında sırası ile %13,17 ve %2,23 oranında daha yüksek başarı sağlamıştır. Varyasyonlu otomatik kodlayıcı ve Gauss karışımı tabanlı yarı denetimli kumaş hata tespit yöntemi olan VGS [141] modeli ile AITEEX veri kümesinde %81,0 F1 skoruna ulaşılmışken, önerilen DSUB-EUNet modeli ile %90,39 F1 skoruna ulaşılmıştır. UNet'in

atlama bağlantısına geliştirilen SU tabanlı dikkat modeli ekleyerek geliştirilen CU-Net [88] yaklaşımı %92,7 duyarlılık başarısına ulaşmıştır.

Sonuç olarak, MT, DAGM ve AITEX veri kümeleri için önceki çalışmalara kıyasla önerilen DSUB-EUNet ağı daha yüksek performanslar sağlamıştır.

4.2.3.2 DSUB-EUNet mimarisinin bileşen analizi

Önerilen DSUB-EUNet, birçok bileşenden oluşan bir ağ mimarisidir. Bu bölümde önerilen ağın bileşenlerini ve başarıya olan katkısını analiz etmek için bir takım deneysel çalışmalar yapılmıştır. İlk yapılan deneysel çalışmada önerilen modelin bileşenleri analiz edilmiştir. İkinci deneysel çalışmada DSUB modeli ile KSMU bloğu karşılaştırılmıştır.

Çizelge 4.10'da gösterildiği gibi, önerilen ağın önceden eğitilmiş EfficientNet, DSUB ve ÇÖB bileşenleri birlikte (Model 4) güçlü bir yapı göstermiş ve en yüksek sonuçları üretmiştir. Model 3 ve 4 karşılaştırıldığında, ÇÖB bloğuna dayalı Model 4, Model 3'e göre MT, DAGM ve AITEX veri kümelerinde sırasıyla %1,34, %3,14 ve %0,71 oranında performans artışı sağlamıştır. Ayrıca MT ve DAGM veri kümeleri için önceden eğitilmiş ağırlık kullanılmayan Model 1 ile en düşük sonuçlar elde edilirken, AITEX veri kümesi için en düşük sonucu önceden eğitilmiş EfficientNet tabanlı Model 2 ile elde edilmiştir. Öte yandan, DSUB bloğuna dayalı Model 3, MT, DAGM ve AITEX veri kümeleri için Model 2'ye göre sırasıyla %8,19, %0,45 ve %1,05 mIoU performans artışı sağlamıştır.

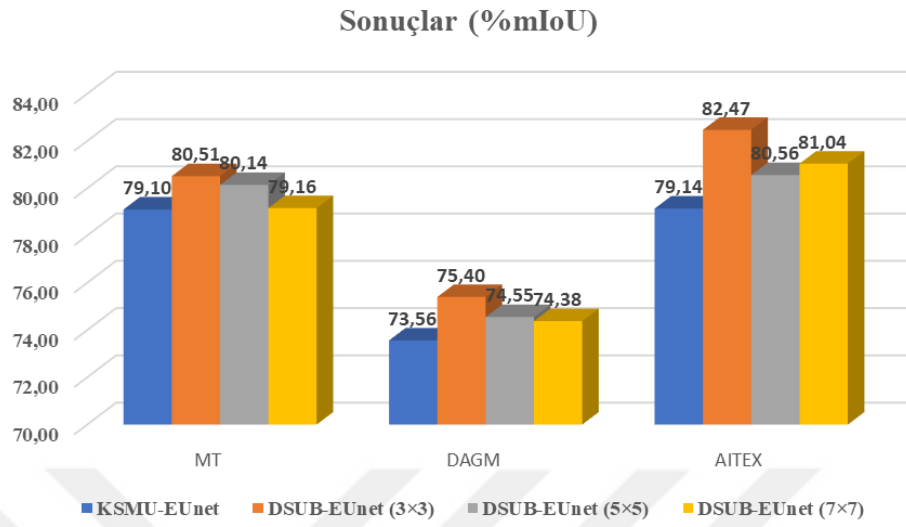
Çizelge 4.10: DSUB-EUNet bileşenlerinin analizi.

Modeller	Bileşenler			mIoU		
	Önceden eğitilmiş EfficientNet	DSUB	ÇÖB	MT	DAGM	AITEX
Model 1	X	X	X	66,58	64,95	81,30
Model 2	✓	X	X	70,98	71,62	80,71
Model 3	✓	✓	X	79,17	72,07	81,76
Model 4	✓	✓	✓	80,51	75,21	82,47

Sonuç olarak önerilen mimaride yer alan DSUB ve ÇÖB yapıları, hata tespiti için kritik özellikleri güçlendirmiş ve bunları son katmana aktararak yüksek bir başarı elde etmiştir.

Önerilen DSUB modeli KSMU yapısını temel alarak geliştirilmiştir. DSUB ve KSMU yapılarının daha iyi karşılaştırılabilmesi için birtakım analizler yapılmıştır. Bu analizler için önerilen DSUB-EUNet modeline benzer KSMU tabanlı EfficientNet-UNet (KSMU-EUNet) modeli geliştirilmiştir. DSUB-EUNet ve KSMU-EUNet mimarileri aynı deneysel çalışmalar uygulanarak Şekil 4.14'te verilen sonuçlar elde edilmiştir. Burada verilen DSUB-EUNet

modellerindeki ($c \times c$) parametresi, DSUB bloğunda kullanılan derinlemesine evrişim katmanının çekirdeğidir.

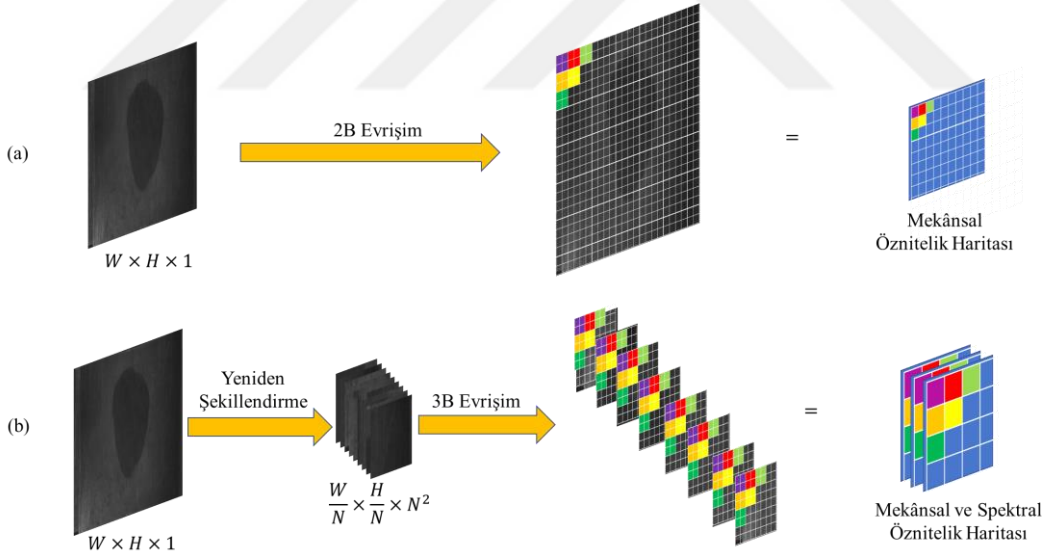


Şekil 4.14: DSUB-EUNet ve KSMU-EUNet test sonuçları.

Şekil 4.14'teki sonuçlara göre önerilen DSUB-EUNet modeli, KSMU-EUNet modelinden daha iyi performans göstermiştir. MT veri kümesinde, DSUB-EUNet (3x3) modeli ile %80,51 mIoU elde edilirken, KSMU-EUNet modeli ile %79,10 mIoU elde edilmiştir. DAGM veri kümesinde DSUB-EUNet (3x3) modeli, KSMU-EUNet modeline göre %1,84 oranında performans artışı sağlamıştır. Benzer şekilde AITEX veri kümesinde, DSUB-EUNet (3x3) modeli ile %3,33 mIoU oranında performans artışı sağlamıştır. Sonuç olarak 3x3 çekirdek boyutunu kullanan DSUB-EUNet modeli kullanılarak MT, DAGM ve AITEX veri kümelerinde en yüksek başarılar elde edilmiştir.

4.3 Uygulama 3: Çok Boyutlu Öznitelik Çıkarma Tabanlı Derin Kodlayıcı-Kod Çözücü Ağ Mimarisi

Kumaş, ahşap ve metal gibi yüzeylerde bulunan bazı hatalar görüntü boyunca farklı doku öznitelikleri gösterebilmektedir. Bundan dolayı görüntüdeki iki uzak nokta arasındaki kenar, renk ve doku öznitelikleri hata tespiti için önemli olacaktır. Fakat ESA mimarilerinde, evrişim yerelliği (locality of the convolution) [106, 144] nedeniyle ilk katmanlarda, farklı noktalar arasındaki ilişki ele alınamamaktadır. Farklı noktalar arasındaki ilişki daha uzak (derin) katmanlarda değerlendirilmektedir. Daha uzak veya daha derin katmanlara doğru, evrişim, ReLU ve havuzlama katmanlarından [7, 67] dolayı girdinin temel renk, doku ve kenar gibi mekânsal öznitelikleri kaybolmaktadır. Bu problemi ele alarak mevcut çalışmada, daha kapsamlı ve daha etkili mekânsal-spektral öznitelikler çıkarmak için 3B Spektral ve Mekânsal Öznitelikler Çıkarma (3BSM) modeli geliştirilmiştir. 3BSM modeli ile 2B evrişim işlemi arasındaki temel fark Şekil 4.15'te gösterilmiştir. 3BSM modelinde öncelikle görüntü yeniden şekillendirilerek 3 boyutlu görüntü formatına getirilmiştir. Daha sonra, 3 Boyutlu (3B) evrişim işlemi kullanılarak geniş bir yelpazede mekânsal ve spektral bilgi içeren bir öznitelik haritası elde edilmiştir.



Şekil 4.15: (a) 2B evrişim ve (b) 3BSM modülü.

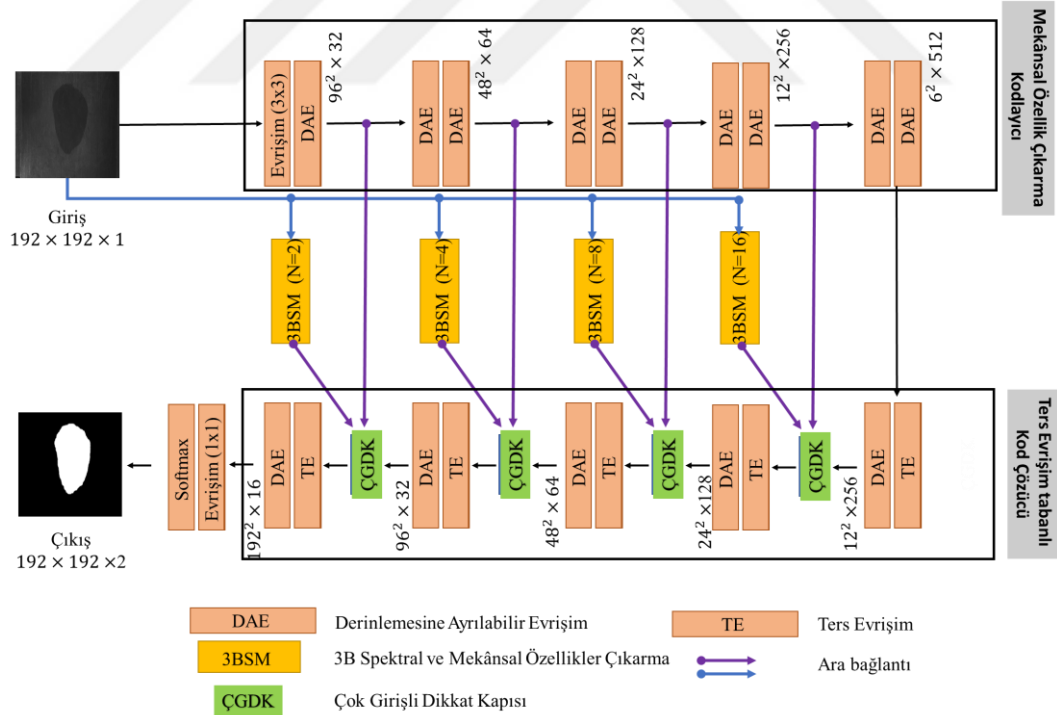
Bu uygulamada önerilen 3BSM modelini, yüzey hata tespitine uygulamak için küçük veri kümelerinde yüksek performans sağlayabilecek basit fakat etkili bir kodlayıcı-kod çözücü mimarisi tasarlanmıştır. Çok Boyutlu Öznitelik Çıkarma tabanlı Derin Kodlayıcı-Kod Çözücü Ağ (Multi-dimensional Feature Extraction based Deep Encoder-Decoder Network

(MFE-DEDNet)) mimarisi olarak adlandırılan bu ağ mimarisi sadece 5,1 milyon parametreye sahiptir. Önerilen modelin detayları aşağıda verilmiştir.

4.3.1 Önerilen model

MFE-DEDNet modelinin genel yapısı Şekil 4.16’da verilmiştir. MFE-DEDNet ağı, kodlayıcı-kod çözücü ağ stiline sahiptir. Bu ağ mimarisinin başlıca özellikleri şunlardır:

- MFE-DEDNet ağının kodlayıcı kısmında, giriş görüntüsünden iki farklı türde derin öznetelik haritası elde edilmiştir. Birincisi, DAE katmanlarından oluşan Mekânsal öznetelik çıkarma (MÖÇ) modülünden elde edilen öznetelik haritalarıdır. İkincisi, 3BSM modülünden elde edilen spektral ve mekânsal bilgileri içeren öznetelik haritasıdır.
- MFE-DEDNet ağının kod çözücü bölümü DAE ve Ters Evrişim (TE, ConvTranspose) katmanlarından oluşmaktadır. Ayrıca bu bölümde Çok Girişli Dikkat Kapısı (ÇGDK) kullanılmıştır. ÇGDK blokları elde edilen düşük seviyeli öznetelik haritalarını, yüksek seviyeli öznetelik haritalarını ve spektral/mekânsal öznetelik haritalarını etkili bir şekilde birleştirmesini sağlamıştır.



Şekil 4.16: MFE-DEDNet ağının genel yapısı.

Önerilen MFE-DEDNet ağ modelinin bileşenleri aşağıda alt başlıklar halinde detaylandırılmıştır.

4.3.1.1 Çok boyutlu mekânsal öznitelik çıkarma

ESA tabanlı mimarilerde, görüntülerdeki öznitelikleri elde etmek için birden çok ardışık Evrişim, ReLU ve Batch Normalizasyon katmanları kullanılmıştır. Bununla birlikte birçok çalışmada katman sayısı artırılarak daha derin ve anlamsal öznitelikler elde edilmeye çalışılmıştır [76, 82, 88]. ESA'nın çok fazla katman içermesi, kaybolan gradyan sorununa neden olmaktadır. Ayrıca kullanılan her evrişim (Conv) işleminde öğrenilecek parametre sayısı arttığı için ağ eğitiminde büyük veri kümelerine ihtiyaç duyulmaktadır. Ancak, hata tespitinde genellikle sınırlı sayıda örnek içeren veri kümeleri vardır [55, 127, 145]. Bu problemi ele alarak, önerilen ağı kodlayıcı bölümü için düşük parametreye ve katmana sahip Mekânsal Öznitelik Çıkarma (MÖÇ, Spatial Feature Extraction) ağı geliştirilmiştir. Önerilen MÖÇ modelinde VGGNet [76], FPNNet [65], PGANet [7] ve Unet mimarilerinde kullanılan standart evrişim katmanları yerine Derinlemesine Ayrılabilir Evrişim (DAE) [95] katmanları kullanılmıştır. Önerilen MÖÇ ağının katman ve hiperparametreleri Çizelge 4.11'de verilmiştir.

Çizelge 4.11: MÖÇ ağının katmanları ve her katmanda kullanılan hiperparametreler.

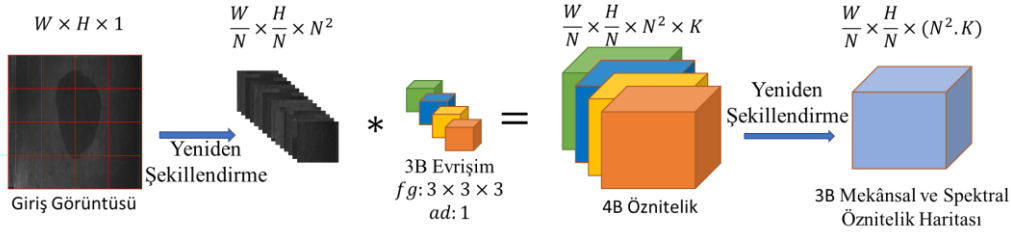
Seviyeler	Katmanlar	Adım	Filtre	Çıktı
Görüntü	Giriş	-	-	192x192x1
Seviye 1	Evrişim+BN+ReLU	2x2	32	96x96x32
	DAE	1x1	32	96x96x32
Seviye 2	DAE	2x2	64	48x48x64
	DAE	1x1	64	48x48x64
Seviye 3	DAE	2x2	128	24x24x128
	DAE	1x1	128	24x24x128
Seviye 4	DAE	2x2	256	12x12x256
	DAE	1x1	256	12x12x256
Seviye 5	DAE	2x2	512	6x6x512
	DAE	1x1	512	6x6x512

MÖÇ ağı toplam beş seviyeden oluşmaktadır. DAE katmanı her seviyede iki kez uygulanmıştır. MÖÇ modülünün öznitelik haritasının boyutunu küçültmek için her seviyenin başında DAE'deki DE işleminin adım genişlik değeri 2x2 olarak alınmıştır. Böylece, öznitelik haritasının boyutu herhangi bir havuzlama katmanı kullanılmadan azaltılmıştır. Öte yanda DAE katman içerisinde bulunan her DE ve NE katmanından sonra Batch normalizasyon ve ReLU katmanları kullanılmıştır.

4.3.1.2 3B spektral ve mekânsal öznitelikler çıkarma

3B Spektral ve Mekânsal Öznitelikler Çıkarma (3BSM, 3D Spectral and Spatial Features Extract) modülünün amacı hem yakın hem de uzak pikselleri ilişkilendirerek etkin spektral

ve mekânsal (uzaysal) öznitelikler elde etmektir. 3BSM modülünde öncelikle görüntü yeniden şekillendirilerek 3 boyutlu görüntüye dönüştürülür. Ortaya çıkan görüntü, Multispektral [146] veya 3B biyomedikal [93] görüntülere büyük ölçüde benzerlik gösterir. Ardından, elde edilen 3B görüntüye bir 3B evrişim işlemi uygulanarak spektral ve mekânsal öznitelikler elde edilir (Şekil 4.17). Bu sayede görüntüdeki farklı noktalardan spektral ve mekânsal öznitelikler elde edilir.



Şekil 4.17: 3BSM modülünün işlem adımları.

Şekil 4.17’de gösterilen W ve H değerleri, sırasıyla giriş görüntüsünün genişlik ve yükseklik boyutlarını temsil eder. N^2 değeri, görüntünün kaç parçaya bölüldüğünü göstermiştir. Görüntü parçalamanın sonucu olarak, $\frac{W}{N} \times \frac{H}{N}$ boyutunda N^2 adet görüntü parçası elde edilmiştir. Daha sonra bu görüntüler üçüncü eksen boyunca birleştirilmiş ve $\frac{W}{N} \times \frac{H}{N} \times N^2$ boyutlu bir 3B görüntü elde edilmiştir. Elde edilen bu görüntü, 3B evrişim (çb:3x3x3; ad:1x1x1 fs:K) işleminden geçirilmiştir. Evrişim işlemi sonucunda 4 boyutlu bir öznitelik küpü $\frac{W}{N} \times \frac{H}{N} \times N^2 \times K$ elde edilmiştir. Son olarak, 4 boyutlu öznitelik küpünün 3. ve 4. boyutları birleştirilmiştir. Sonuç olarak 3 boyutlu bir öznitelik haritası $\frac{W}{N} \times \frac{H}{N} \times (N^2.K)$ elde edilmiştir.

MFE-DEDNet mimarisinde MÖÇ ağına paralel olarak her seviye için bir 3BSM kullanılmıştır. 3BSM modülünde kullanılan N değerini ve her seviye için kullanılan filtre sayısı Çizelge 4.12’de verilmiştir. Önerilen modelde 3B evrişimden kaynaklı yüksek hesaplama maliyetini önlemek için 3BSM modülünde kullanılan filtre sayısı, N değeri ile ters orantılı olarak 8, 4, 2 ve 1 değerleri kullanılmıştır. Bu sayede 3B evrişimden kaynaklanan yüksek işlem maliyetinin önüne geçilmiştir.

Çizelge 4.12: MFE-DEDNet ağında kullanılan 3BSM modülünün hiper parametreleri.

Seviyeler	Görüntü Boyutu	Katmanlar	N	Filtre	Çıktı
Seviye 1	96x96x4	3BSM	2	8	96x96x32
Seviye 2	48x48x16	3BSM	4	4	48x48x64
Seviye 3	24x24x64	3BSM	8	2	24x24x128
Seviye 4	12x12x256	3BSM	16	1	12x12x256

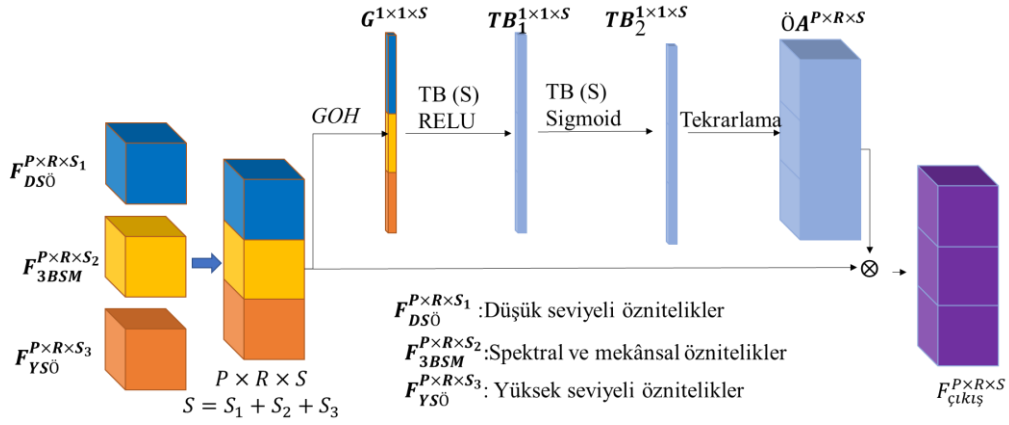
4.3.1.3 Çok girişli dikkat kapısı

MFE-DEDNet mimarisinin kod çözücü biriminde, elde edilen üç farklı öznitelik haritası birleştirilerek kullanılmıştır. Bu öznitelik haritaları şunlardır:

- 1) 3BSM modülünden elde edilen spektral ve mekânsal öznitelikler,
- 2) Atlamalı bağlantılarıyla aktarılan düşük seviyeli öznitelikler,
- 3) Hiyerarşik katmanlar yolu ile elde edilen yüksek seviyeli özniteliklerdir.

Her öznitelik haritası kendi içinde anlamsal (semantic), mekânsal (uzamsal) ve spektral (3BSM'nin 3B evrişim çıkışlarında [147]) bilgiler taşımaktadır. Mekânsal ayrıntılar renk, kenar ve doku gibi bilgileri içerirken, anlamsal ayrıntılar sınıflandırma yapma konusunda yetenekli ancak konum ve şekil bakımından zayıf olan bağlam bilgilerini içermektedir [7, 148]. Öte yandan, spektral bilgiler 3B evrişim yoluyla giriş görüntüsünün farklı noktaları arasındaki mekânsal ayrıntıları içermektedir [80, 147]. Bu öznitelikleri elde etme yöntemleri farklı olduğu için farklı içeriklere sahiptirler. Bundan dolayı, bu çalışmada, bu öznitelik haritalarını doğrudan birleştirmek yerine, 3 girişli bir dikkat kapısı kullanılmıştır. Çok Girişli Dikkat Kapısı (ÇGDK, Multi Input Attention Gate) olarak adlandırılan bu model temel olarak sıkma ve uyarılma (SU) [105] modeline dayanmaktadır.

ÇGDK modelinde öncelikle farklı öznitelik haritaları kanal ekseninde birleştirilmiştir. Daha sonra birleştirilen öznitelikler için SU yapısı kullanılarak ağırlık vektörü elde edilmiştir. Son olarak bu ağırlık vektörü ile özniteliklerdeki önemli detaylar ortaya çıkarılmıştır. ÇGDK yapısının detayı Şekil 4.18'de gösterilmiştir.



Şekil 4.18: Çok Girişli Dikkat Kapısı.

Şekil 4.18’de verilen ÇGDK yaklaşımının sözde kodu Algoritma 4.2’de verilmiştir.

Algoritma 4.2: Önerilen ÇGDK yaklaşımının sözde kodu.

Girdi: $f_{DSÖ}^{P \times R \times S_1}, f_{3BSM}^{P \times R \times S_2}, f_{YSÖ}^{P \times R \times S_3}$ öznitelikler	
Çıktı: $F_{çıkış}^{P \times R \times S}$ öznitelik haritası	
1:	$F^{P \times R \times S} = \text{concat}(f_{DSÖ}^{P \times R \times S_1}, f_{3BSM}^{P \times R \times S_2}, f_{YSÖ}^{P \times R \times S_3}, \text{axis} = 3)$
2:	$G^{1 \times 1 \times S} = \text{GOH}(F)$, GOH global ortalama havuzlama katmanını
3:	$tb_1^{1 \times 1 \times S} = \text{ReLU}(w * G^{1 \times 1 \times S} + b)$,
4:	$tb_2^{1 \times 1 \times S} = \text{sigmoid}(w * tb_1 + b)$
5:	$ÖA_{map}^{P \times R \times S} = \text{Repeat}(tb_2^{1 \times 1 \times S})$, ÖA öznitelik ağırlıkları
6:	$F_{çıkış}^{P \times R \times S} = ÖA_{map}^{P \times R \times S} \otimes F^{P \times R \times S}$

Algoritma 4.2’de gösterildiği gibi, ÇGDK modülünde sırası ile şu işlemler uygulanmıştır:

- Birinci adımda ÇGDK modülünün aynı yükseklik, genişlik ve derinliğe sahip üç farklı öznitelik haritası üçüncü eksenlerinde ($axis = 3$) birleştirilmiştir ($concat$).
- İkinci adımda, birleştirilmiş öznitelik haritasına ($F^{P \times R \times S}$) global ortalama havuzlama ($GOH(\cdot)$, GlobalAVGPoolin) katmanı uygulanmıştır. İşlem sonucunda $G^{1 \times 1 \times S}$ sinyali elde edilmiştir.
- Üçüncü adımda elde edilen sinyal Şekil 4.18’de verildiği gibi tam bağlı (TB) katmanlardan geçirilmiştir.
- Dördüncü adımda sigmoid fonksiyonundan geçirilerek 0 ile 1 arasında değişen bir ağırlık matrisi ($1 \times 1 \times S$) elde edilmiştir.
- Beşinci adımda ağırlık haritası girdi ile aynı boyuta getirilmiştir.
- Son adımda, elde edilen öznitelik ağırlıkları (ÖA), $P \times R \times S$ girdi öznitelik haritası ile eleman bazında çarpma işleminden geçirilmiştir.

Sonuç olarak üç farklı öznitelik haritasını ağırlıklandırma yoluyla birleştiren bir öznitelik haritası ($F_{\text{çıkış}_\text{öznitelik}_\text{haritası}}^{P \times R \times S}$) elde edilmiştir.

4.3.1.4 Ters evrişim tabanlı kod çözücü

MFE-DEDNet mimarisinin kodlayıcı bölümündeki MÖÇ ve 3BSM modülleri sayesinde güçlü öznitelik haritaları elde edilmiştir. MFE-DEDNet ağının kod çözücü bölümünde Ters Evrişim Tabanlı Kod Çözücü (TEKÇ, Convolution Transpose Base Decoder) ağı ile hata tespit haritası elde edilmiştir. Çizelge 4.13'te TEKÇ ağındaki katmanlar ve parametreler verilmiştir.

Çizelge 4.13: TEKÇ ağındaki katmanları ve hiper parametreleri.

Kod çözücü Seviyesi	Katmanlar	Adım	Filtre Sayısı	Çıktı Boyutu
Seviye 1	TE+BN+ReLU	2x2	256	12x12x256
	ÇGDK	-	-	12x12x768
	DAE	1x1	256	12x12x256
Seviye 2	TE+BN+ReLU	2x2	128	24x24x128
	ÇGDK	-	-	24x24x384
	DAE	1x1	128	24x24x128
Seviye 3	TE+BN+ReLU	2x2	64	48x48x64
	ÇGDK	-	-	48x48x192
	DAE	1x1	64	48x48x64
Seviye 4	TE+BN+ReLU	2x2	32	96x96x32
	ÇGDK	-	-	96x96x160
	DAE	1x1	32	96x96x32
Çıkış Seviyesi	TE + BN+ReLU	2x2	16	192x192x16
	DAE	1x1	16	192x192x16
	Evrişim	2x2	1	192x192x2
	softmax	-	-	192x192x2

TEKÇ ağı beş seviyeden oluşmaktadır. Her seviye başında, genişlik ve yükseklik boyutunu arttırmak için TE katmanı kullanılmıştır. Daha sonra, önceki katmanlarda elde edilen 3 farklı öznitelik haritası ÇGDK katmanına aktarılmıştır. Seviyenin sonunda, ÇGDK ile birleştirilmiş öznitelik haritası DAE katmanından geçirilmiştir.

Yukarıdaki işlemler TEKÇ ağındaki ilk 4 seviyesinde benzer olarak tekrarlanmıştır. TEKÇ ağındaki son seviyesinde (Çıkış Seviyesi), hata tespit sonucu için öznitelik haritası TE, BN ve ReLU katmanlarından geçirilmiştir. Daha sonra bir DAE katmanı uygulanarak nihai öznitelik haritası elde edilmiştir. Ortaya çıkan öznitelik haritası, görüntü ile aynı boyuttadır ve 16 kanala sahiptir. Son olarak, piksel seviyesinde hata tespiti için, bu öznitelik haritası evrişim ve softmax katmanlarından geçirilmiştir. Bu işlem sonucunda $192 \times 192 \times 2$

boyutlu ağ çıktısı elde edilmiştir. Çıktının birinci ve ikinci kanalları sırası ile arka plan ve hatalı pikselleri temsil etmektedir.

4.3.1.5 Model eğitimi

Yukarıda ayrıntıları verilen MÖÇ, 3BSM, ÇGDK ve TEKÇ modüllerinin birleştirilmesiyle MFE-DEDNet ağı geliştirilmiştir. Önerilen MFE-DEDNet, toplam 5,1 milyon parametreye sahip uçtan uca bir ağıdır. Önerilen ağ mimarisi birçok bileşenden oluşmasına rağmen yaklaşık 31 milyon parametre içeren UNet [67] mimarisinden altı kat daha az parametreye sahiptir. Bu sayede diğer birçok ağ mimarisinin aksine deneysel çalışmalarda belirtildiği gibi az sayıda örnek içeren veri kümeleri ile eğitilebilmiştir. MFE-DEDNet ağı, hata tespitini piksel bazında bir segmentasyon problemi olarak ele almaktadır. Bu nedenle, ağın eğitiminde İkili Çapraz Entropi (Binary Cross-Entropy, Denklem 4.3) kayıp fonksiyonu kullanılmıştır.

4.3.2 Deneysel çalışmalar ve sonuçları

Önerilen MFE-DEDNet mimarisinin performansını değerlendirmek için MT ve MV Tec veri kümeleri üzerinde kapsamlı deneysel çalışmalar ve analizler yapılmıştır. Ayrıca Spektral ve Mekânsal Öznitelikler Çıkarma (3BSM) ve ÇGDK modüllerinin daha detaylı analizi için sadece MÖÇ (kodlayıcı) ve TEKÇ (kod çözücü) ağlarından oluşan SFEDNet mimarisi geliştirilmiştir. Önerilen SFEDNet ve MFE-DEDNet mimarileri UNet ve Mobile-UNet ağları ile karşılaştırılmıştır. Bunun yanı sıra her veri kümesi için daha önce yapılmış çalışmaların sonuçları ele alınmış ve değerlendirilmiştir.

4.3.2.1 Veri kümeleri ve uygulama detayları

MFE-DEDNet, ağın detaylı değerlendirilmesi için literatürde yaygın olarak kullanılan MT ve MV Tec veri kümesi ile eğitilmiş ve test edilmiştir. Deneysel çalışmalarda MT veri kümesinin 392 hatalı görüntüsü kullanılmıştır. Bunun yanı sıra MV Tec veri kümesindeki 15 sınıfa ayrı ayrı uygulanmış ve sonuçlar kaydedilmiştir. Eğitim setindeki görüntülere döndürme (90° , 180° ve 270°), kırpma, bulanıklaştırma ve çevirme gibi veri arttırma yöntemlerini kullanarak veri kümesindeki örnek sayısı 8 katına çıkartılmıştır.

Deneysel çalışmalarda, önerilen model 4-kat çapraz doğrulama [149] yöntemi kullanılarak eğitilmiş ve test edilmiştir. Önerilen ağların eğitimi için öğrenme oranı, batch boyutu ve epoch sayısı sırasıyla 0,01, 16 ve 100 olarak belirlenmiştir. Ağ parametrelerinin optimizasyonu için Adam optimizasyon yöntemi kullanılmıştır.

4.3.2.2 MT veri kümesi

MT veri kümesi kullanılarak elde edilen sonuçlar Çizelge 4.14'te verilmiştir. Verilen sonuçlarda, önerilen MFE-DEDNet yöntemi ile en yüksek skorlar (%80,01 mIoU ve %88,86 f1-skoru) elde edilmiştir. Elde edilen sonuçlarda ikinci sırada yer alan SFEDNet ağı, kullanılan DAE katmanları ile hata tespiti için güçlü bir kodlayıcı-kod çözücü yapısı sunmuştur. DAE katmanları ile oluşturulan SFEDNet ve MFE-DEDNet modellerinin performansları kıyaslandığında, MFE-DEDNet modelinde bulunan 3BSM ve MÖÇ modüllerinin başarıya olumlu etki ettiği görülmüştür. Öte yandan Mobile-Unet ve Unet mimarileri önerilen modellerden daha düşük sonuçlar vermiştir. Çizelge 4.14'te verilen sonuçlara göre önerilen MFE-DEDNet modeli literatürde bulunan diğer çalışmalara göre daha yüksek başarı sağlamıştır. Hata tespitinde örnek sayısının az olması nedeniyle PGA-Net ve DFF ağlarında önceden eğitilmiş ağ mimarileri kullanılmıştır. Buna kıyasla 5,1 milyon parametre içeren MFE-DEDNet mimarisi, önceden eğitilmiş ağ mimarilerinden daha yüksek skorlar sağlamıştır.

Çizelge 4.14: MT veri kümesi ile elde edilen sayısal sonuçlar.

Referanslar	Yöntemler	mIoU	F1-skor	AUC
MCuePushU [69]	UNet, YSA ve temel görüntü işleme	-	-	96,6
PGA-Net[7]	VGG16, FPNet ve Dikkat Kapısı	71,31		
DifferNet [73]	ESA, akış algoritmasını normalleştirme	-	-	97,7
DPAE[140]	otomatik kodlayıcı		78,0	91,0
UISDI[150]	ESA, temel görüntü işleme	-	76,9	-
DFF[41]	VGG16, Kodlayıcı-Kod çözücü ve Dikkat Kapısı	73,7	82,2	-
Mobil-UNet [86]	Mobilenet ve UNet	61,19	70,06	88,98
UNet [151]	UNet	68,87	81,50	97,80
Önerilen SFEDNet	Önerilen Kodlayıcı (MÖÇ) ve Kod Çözücü (TEKÇ)	77,33	87,20	98,38
Önerilen MFE-DEDNet	Önerilen Kodlayıcı (MÖÇ), Kod çözücü (TEKÇ), 3BSM ve Dikkat Kapısı (ÇGDK)	80,01	88,86	99,07

MT veri kümesinden alınan bazı örnekler ve yöntemlerin tahmin sonuçları Şekil 4.19'da verilmiştir. Sonuçların sayısal olarak karşılaştırılması için elde edilen tahmin görüntülerinin altında hesaplanan IoU değeri verilmiştir. Verilen örneklerde görüldüğü üzere yüzey hataları, MFE-DEDNet ve SFEDNet ağları ile daha iyi tespit edilmiştir. Şekil 4.19'daki

örnek 2 ve 3'te küçük ve tespit edilmesi zor yüzey hataları içermektedir. Bu örneklerde önerilen MFE-DEDNet modeli kullanılarak yüksek doğrulukta yüzey hatası tespit edilmiştir. Öte yandan SFEDNet modeli kullanılarak sadece örnek 2'de başarılı sonuçlar vermiştir. 5. örnekte verilen yüzey hatası arka plan ile benzerlik göstermektedir. Bu örnekte önerilen MFE-DEDNet ve UNet modelleri ile yüzey hatası başarılı bir şekilde tespit edilirken diğer yöntemler ile yüzey hatası kısmen tespit edilmiştir.

Örnek	Gerçek Değer	UNet	Mobile-Unet	SFEDNet	MFE-DEDNet
Örn. 1					
		iou:74.79	iou:89.51	iou:95.27	iou:94.03
Örn. 2					
		iou:14.91	iou:54.71	iou:76.78	iou:91.07
Örn. 3					
		iou:50.44	iou:53.92	iou:6.11	iou:71.98
Örn. 4					
		iou:40.16	iou:29.43	iou:71.12	iou:92.33
Örn. 5					
		iou:76.85	iou:62.02	iou:57.62	iou:84.88

Şekil 4.19: MT veri kümesinden görsel sonuçlar.

4.3.2.3 MVTEC veri kümesi




















MVTEC veri kümesi 15 farklı sınıf ve her sınıf içerisinde farklı hata türleri ile oluşturulmuş en kapsamlı veri kümelerinden biridir. MVTEC için literatürde bulunan ve önerilen modellerin performans sonuçları Çizelge 4.15'te verilmiştir. Bu veri kümesinde her sınıf için 100-400 arasında görüntü ve bu görüntülerden küçük bir kısmı yüzey hatası içermektedir. Bu nedenle, geleneksel ESA mimarileri için zor bir veri kümesidir. Önerilen MFE-DEDNet ağı, basit yapısı sayesinde bu veri kümesinde olağanüstü bir başarı sağlamıştır. Çizelge 4.15'te verildiği gibi birçok sınıf için MFE-DEDNet ile en yüksek sonuçlar elde edilmiştir. Öte yandan Unet modeli ile bazı sınıflarda yüksek skorlar sağlarken Mobile-Unet modeli genel olarak düşük sonuçlar üretmiştir.

Çizelge 4.15: MVTEc veri kümesi ile elde edilen sayısal sonuçlar.

Veri kümesi	Metrik	MFE-DEDNet	SFEDNet	UNet [151]	Mobil-UNet [86]	P-net [128]	Trust-MAE [129]	TVE [130]	Differ Net [73]	RİAD [131]
Hah	AUC	98,95	98,55	96,56	95,02	57,00	98,53	78,00	92,90	96,30
	mIoU	64,42	59,57	46,99	40,98	14,00		0,10		
	F1-skoru	78,33	74,25	60,02	51,18					
Izgara	AUC	98,69	98,10	98,35	97,49	98,00	97,45	73,00	84,00	98,80
	mIoU	46,39	49,60	49,55	32,61	59,00		0,02		
	F1-skoru	63,17	66,31	66,09	48,19					
Deri	AUC	99,83	99,61	98,94	99,34	89,00	98,05	95,00	97,10	99,40
	mIoU	78,61	75,00	58,59	62,05	52,00		0,24		
	F1-skoru	87,91	85,63	73,53	76,56					
Fayans	AUC	97,49	98,52	99,31	86,12	97,00	82,48	80,00	99,40	89,10
	mIoU	59,56	63,43	74,20	29,51	23,00		0,23		
	F1-skoru	70,31	76,47	85,16	40,10					
Oduun	AUC	96,48	95,92	93,55	95,68	98,00	92,62	77,00	99,80	85,80
	mIoU	51,01	49,33	33,77	25,17	37,00		0,14		
	F1-skoru	67,02	65,53	46,91	38,48					
Şişe	AUC	99,10	99,63	94,95	97,37	99,00	93,39	87,00	99,00	98,40
	mIoU	75,40	78,06	37,88	47,16	43,00		0,27		
	F1-skoru	85,93	87,65	47,41	58,17					
Kablo	AUC	91,12	91,52	92,38	93,22	70,00	98,53	90,00	95,90	84,20
	mIoU	49,19	46,98	51,78	35,54	16,00		0,18		
	F1-skoru	65,36	62,63	67,83	51,21					
Kapsül	AUC	83,87	91,62	96,06	79,31	84,00	87,42	74,00	86,90	92,80
	mIoU	22,93	26,92	25,09	24,79	64,00		0,11		
	F1-skoru	35,13	41,16	37,58	37,20					
Fındık	AUC	99,62	99,76	98,70	98,76	97,00	98,51	98,00	99,30	96,10
	mIoU	79,31	75,86	64,24	75,60	66,00		0,44		
	F1-skoru	88,38	86,23	77,98	85,55					
Metal somun	AUC	99,04	98,56	96,66	93,47	79,00	91,76	94,00	96,10	92,50
	mIoU	82,06	82,03	55,92	52,40	24,00		0,49		
	F1-skoru	90,06	89,97	65,31	65,93					
Hap	AUC	98,95	94,50	92,57	84,22	97,00	89,90	83,00	88,80	95,70
	mIoU	60,34	37,03	7,81	23,13	58,00		0,18		
	F1-skoru	73,63	49,84	13,45	27,84					
Vida	AUC	86,38	86,62	88,94	93,12	100,0	97,63	97,00	96,30	98,80
	mIoU	20,00	35,38	19,55	15,19	32,00		0,17		
	F1-skoru	32,84	52,14	31,71	24,99					
Diş fırçası	AUC	93,00	89,45	64,64	91,34	99,00	98,10	94,00	98,60	98,90
	mIoU	46,96	33,94	20,76	29,21	63,00		0,14		
	F1-skoru	63,67	47,60	28,47	41,28					
Transistör	AUC	98,03	91,03	70,25	84,14	84,00	92,72	93,00	91,10	87,70
	mIoU	52,61	21,39	6,86	27,85	24,00		0,30		
	F1-skoru	68,12	34,48	11,97	39,38					
Fermuar	AUC	99,28	98,79	99,00	98,40	90,00	97,76	78,00	95,10	97,80
	mIoU	63,34	65,78	47,80	57,22	34,00		0,06		
	F1-skoru	77,55	79,32	60,34	72,09					
Ortalama	AUC	95,99	95,48	92,06	92,47	89,20	94,32	86,07	94,69	94,15
	mIoU	56,81	53,37	40,05	38,56	40,60		0,20		
	F1-skoru	69,83	66,61	51,58	50,54					

Çizelge 4.15'te verilen P-net [128], Trust-MAE [129], TVE [130], DifferNet [73] ve RIAD [131] gibi otomatik kodlayıcı modelleri, görüntü yeniden oluşturma tabanlı yaklaşımlardır. MVTec veri kümesinde bulunan fındık, hap, transistör, fermuar gibi bazı sınıflarda küçük ve ince detaylara sahip yüzey hataları içermektedir. Ayrıca bu yüzeylerde oluşan hatalar arka plan ile benzerlik göstermektedir. Bundan dolayı bu sınıflarda otomatik kodlayıcılar ile düşük performanslar elde edilmiştir. Buna kıyasla öğrenme tabanlı MFE-DEDNet mimarisi bu sınıflarda daha başarılı bir performans göstermiştir.

MVTec veri kümesinden alınan bazı örnek görüntüler ve modellerin tahmin sonuçları Şekil 4.20'de verilmiştir. 1 ve 4 arasındaki örnek görüntülerde basit ve belirgin yüzey hataları vardır. Ancak 8, 9, ve 10'daki gibi bazı örneklerde yüzey hataları oldukça küçüktür. Bunun yanı sıra 11 ve 20 arasındaki örnekler arka plan ile benzerlik gösteren yüzey hataları içerirken, 21 ve 24 arasındaki örnekler çok karmaşık bir arka plana sahiptir. Şekil 4.20'de verilen görsel sonuçlarda görüleceği üzere, MFE-DEDNet ağı ile bu problemler karşısında başarılı sonuçlar sağlamıştır. Bunun yanı sıra Mobil-Unet modeli sadece bazı görüntülerde (örnek 2, 7, 11, 19) başarılı olmuşken Unet modeli ise ortalama bir başarı elde etmiştir.

Örnek	Gerçek Değer	UNet	Mobile-UNet	SFEDNet	MFE-DEDNet	Örnek	Gerçek Değer	UNet	Mobile-UNet	SFEDNet	MFE-DEDNet
											
Örn. 1:Fayans		42.28	34.72	70.53	87.8	Örn. 2:Fayans		85.9	72.65	89.03	87.74
											
Örn. 3:Şişe		19.55	23.77	84.12	74.09	Örn. 4:Şişe		19.55	23.77	84.12	74.9
											
Örn. 5: Kablo		50.98	57.07	64.83	59.76	Örn. 6: Kablo		48.9	45.67	51.1	51.3
											
Örn. 7:Halı		70.42	66.24	67.93	77,7	Örn. 8:Halı		50,35	59,11	49,35	61,13
											
Örn. 9:Kapsül		48,48	0,0	0,0	61,33	Örn. 10:Kapsül		40,40	0,0	0,0	61,53
											
Örn. 11:Fermuar		58,42	82,11	86,53	83,36	Örn. 12:Fermuar		55,33	53,71	72,14	78,08
											
Örn. 13:Odun		80,34	78,29	86,4	87,42	Örn. 14:Odun		43,66	0,4	60,13	58,04
											
Örn. 15:Hap		0,0	38,64	55,6	61,13	Örn. 16:Hap		0,0	59,66	5,4	72,19
											
Örn. 17: Metal somun		20,46	44,96	72,01	78,9	Örn. 18: Metal somun		72,24	47,75	82,01	78,59
											
Örn. 19:Fındık		77,3	76,25	77,71	79,87	Örn. 20:Fındık		67,44	64,08	71,23	76,54
											
Örn. 21:Diş fırçası		0,0	2,47	4,88	56,91	Örn. 22:Transistör		7,25	0,0	66,93	
											
Örn. 23:Vida		15,00	0,0	30,71	8,0	Örn. 24:Vida		42,75	0,0	66,26	62,12

Şekil 4.20: MV Tec veri kümesi ile elde edilen görsel sonuçlar.

4.3.3 MFE-DEDNet bileşenlerinin analizi

MFE-DEDNet ağının yapısı ve kullanılan alt modüllerin başarıya olan etkisini araştırmak için bazı deneysel çalışmalar yapılmıştır. İlk yapılan deneysel çalışmada ağ mimarisinin parametre sayısının başarıya olan katkısı analiz edilmiştir. Bunun için MFE-DEDNet mimarisinin alt modeli olan Double_SFEDNet ve SFEDNet modelleri geliştirilmiştir. Bu modellerde sadece MÖÇ (kodlayıcı) ve TEKÇ (kod çözücü) modülleri kullanılmıştır. SFEDNet ağı Çizelge 4.16'da verildiği gibi 3,5 milyon parametreye sahiptir. Bunun yanı

sıra katman ve filtre sayısının iki katına çıkarılmasıyla elde edilen Double_SFEDNet mimarisi 7,3 milyon parametreye sahiptir. İkinci deneysel çalışmada MFE-DEDNet ve SFEDNet sonuçları kıyaslanarak 3BSM ve ÇGDK modüllerinin başarıya olan katkısı araştırılmıştır.

MÖÇ (kodlayıcı) ve TEKÇ (kod çözücü) ağırları, MFE-DEDNet'in omurga ağını oluşturmaktadır. Önerilen ağların oluşturulması sırasında az sayıda katman ve az sayıda parametre içermesi amaçlanmıştır. Çizelge 4.16'da verilen Double_SFEDNet ve SFEDNet mimarilerinin sonuçları karşılaştırıldığında, katman sayılarının artması genel olarak başarıyı düşürmüştür. Bunun temel nedeni, çok az örnek içeren veri kümelerinde, çok katmana sahip ağların yetersiz eğitilmesi ya da aşırı öğrenme problemlerinin oluşmasıdır. Bu problemden yola çıkarak, katman sayısını arttırmak yerine kullanılan ek modüller (3BSM ve ÇGDK) sayesinde önerilen MFE-DEDNet ağı kullanılarak daha yüksek skorlara ulaşılmıştır.

Çizelge 4.16: DEDNet, SFEDNet, Double_SFEDNet mimarilerinin parametre sayısı ve performans sonuçları.

Model	Parametre Sayısı (Milyon)	Fındık	Metal somun	Fayans	Hap	Diş fırçası
MFE-DEDNet	5,1	79,31	82,06	59,56	60,34	52,61
SFEDNet	3,5	75,86	82,23	63,43	37,03	21,39
İki katlı SFEDNet	7,3	75,60	69,40	45,20	30,70	29,76

Çizelge 4.16'daki sonuçlara göre, önerilen MFE-DEDNet ağı ile 3BSM ve ÇGDK modülleri içermeyen SFEDNet ağı ile karşılaştırıldığında metal somun, fındık, hap ve diş fırçası gibi sınıflarda MFE-DEDNet mimarisi daha yüksek başarı sağlamıştır. Bu sonuçlar 3BSM ve ÇGDK modellerinin önemini göstermektedir. Ancak SFEDNet mimarisinin sadece kodlayıcı ve kod çözücü modellerini içermesi, fayans (fayans) gibi homojen ve belirgin yüzey hatalarına sahip örneklerde daha yüksek başarı elde etmesini sağlamıştır.

4.4 Uygulama 4: Dönüştürücü Tabanlı Melez Dikkat Kapısı ve Yeni Bir Ağ Mimarisi

ESA mimarileri, bilgisayar görü alanında üstün performans sağlamış olmasına rağmen, evrişim işleminin yerelliği nedeniyle küresel anlamsal detayların (uzun menzilli bağlamsal bilgisi, long-range contextual information) çıkartılmasında sınırlıdır. Bunun yanı sıra, doğal dil işleme alanında geliştirilen dönüştürücüler (Transformer), öz-dikkat mekanizmalarıyla güçlü küresel semantik (anlamsal) ayrıntıları çıkartmış ve doğal dil işleme alanında büyük başarılar kazanmıştır. Bu başarılardan ilham alan bazı araştırmacılar, bilgisayarlı görü alanında dönüştürücüleri uygulamak için Vision Transformer (ViT) modelini geliştirmiştir [106]. Geliştirilen ViT birçok görüntü sınıflandırma problemine uygulanmış ve yüksek başarılar sağlamıştır [152, 153].

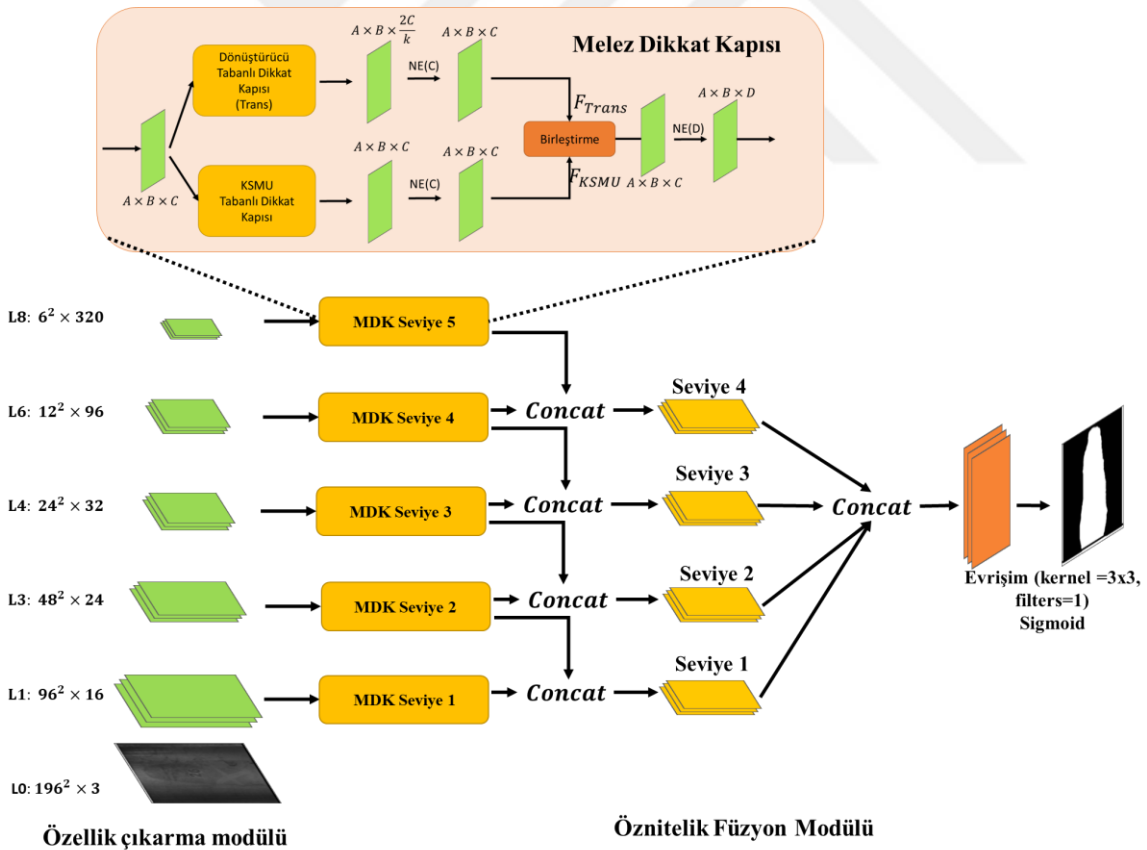
ViT modelinde, öncelikle görüntü parçalanarak tokenleştirme işlemi gerçekleştirilmiştir. Ardından, öz-dikkat mekanizması ile parçalar (belirteçler, token) arasındaki ilişki ele alınarak küresel anlamsal ayrıntılar çıkartılmıştır [111]. [116] nolu çalışmada dönüştürücü yapılarının mekânsal detayları çıkartmada zayıf olduğu açıkça belirtmiştir. Bunun nedeni, dönüştürücülerin girdiyi vektörler olarak ele alması ve sadece küresel anlamsal detayların modellenmesine odaklanmasıdır. Ayrıca, dönüştürücüler elde edilen küresel anlamsal ayrıntıları iki boyutlu çıktıya doğru etkin bir şekilde dönüştüremez [111, 116, 117]. Bu nedenle bazı çalışmalarda [111, 116, 117] mekânsal detayı korumak için ViT blokları, sadece ESA mimarilerinin son katmanlarında (anlamsal ayrıntıların yoğun olduğu öznitelik haritalarında) kullanılmıştır.

Bu çalışmada, hata tespitinde önemli mekânsal (renk, kenar ve doku) ve küresel anlamsal detayları çıkarmak için yeni bir dönüştürücü tabanlı Melez Dikkat Kapısı (MDK, Hybrid Attention Gate) önerilmiştir. Önerilen MDK modeli, Transformer (Trans) ve KSMU modellerinin paralel bağlantısına dayanmaktadır. Trans, hata tespiti için küresel anlamsal ayrıntıları içeren öznitelikleri ortaya çıkartırken, KSMU mekânsal ayrıntıları içeren öznitelikleri ortaya çıkartmaktadır. Deneysel çalışmalarda Trans ve KSMU çıktılarının birleştirilmesinin hata tespiti için önemli öznitelikler ortaya koyduğu ve başarılı sonuçlara ulaşıldığı gözlemlenmiştir. Mevcut çalışmada, hata tespiti için önerilen MDK modeli kullanılarak MDK tabanlı Öznitelik Füzyon Ağı (MDK-ÖFA) geliştirilmiştir. Önerilen MDK-ÖFA modeli aşağıda alt başlıklar ile detaylandırılmıştır.

4.4.1 Önerilen yöntem

MDK-ÖFA ağının ana bileşenleri Şekil 4.21’de verilmiştir. Şekil 4.21’de görüldüğü üzere MDK-ÖFA ağı, öznelik çıkarma modülü, MDK modülü ve öznelik füzyon modülünden oluşmaktadır. Önerilen ağın işlem adımları şunlardır:

- Öznelik çıkarma modülünde, önceden eğitilmiş MobileNetV2 ağı ile giriş görüntüsünden düşük ve yüksek seviyeli başlangıç öznelikleri çıkartılmıştır. Bu modelin çıktısında beş farklı başlangıç öznelik haritası elde edilmiştir.
- MDK modülünde, KSMU ve Trans yapıları ile hata tespiti için başlangıç özneliklerinden önemli mekânsal ve küresel semantik ayrıntıları içeren öznelikler çıkartılmıştır. Ardından, bu öznelik haritaları etkili bir şekilde birleştirilmiştir.
- Öznelik füzyon modülünde, MDK modülünden elde edilen beş farklı öznelik haritası, nihai öznelik haritasını elde etmek için birleştirilmiştir. Öznelik füzyon modülünün son bölümünde, nihai öznelik haritasına evrişim ve sigmoid fonksiyonu uygulanarak hata tespit tahmin haritası elde edilmiştir.



Şekil 4.21: Önerilen MDK tabanlı Öznelik Füzyon Ağı (MDK-ÖFA).

4.4.1.1 Öznitelik çıkarma modülü

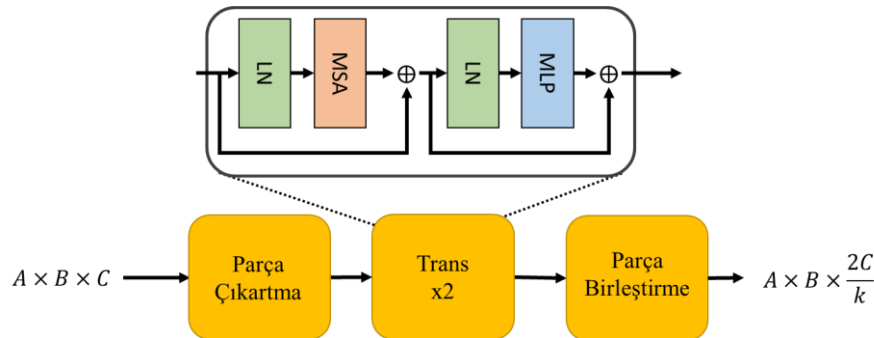
Bu çalışmada, önerilen ağ mimarisinin omurga ağı için önceden eğitilmiş MobileNetV2 [154] mimarisi kullanılmıştır. MobileNetV2 mimarisinin detayları Çizelge 4.17’de verilmiştir. MDK-ÖFA mimarisinin bu modülünde, Trans ve KSMU bloğunda kullanılmak üzere omurga ağından beş farklı başlangıç öznitelik haritası çıkarılmıştır. Başlangıç öznitelik haritaları, sınırlı veri kümesi karşısında dönüştürücülerin daha etkili bir şekilde çalışmasına olanak sağlamıştır. Bu öznitelik haritaları, MobileNetV2 ağının Çizelge 4.17’de verilen L2, L3, L4, L6 ve L8 katman çıktılarıdır.

Çizelge 4.17: MobileNetV2’nin katman yapısı ve parametreleri (t: genişleme değeri, n: tekrarlama sayısı, IRB: NE, DE ve NE katmanları).

Katmanlar	Operasyon	t	fs	n	ag
L0	Evrişim+BN+ReLU6	-	32	1	2
L1	IRB	1	16	1	1
L2	IRB	6	24	2	2
L3	IRB	6	32	3	2
L4	IRB	6	64	4	2
L5	IRB	6	96	3	1
L6	IRB	6	160	3	2
L7	IRB	6	320	1	1
L8	Evrişim+BN+ReLU6	-	1280	1	1

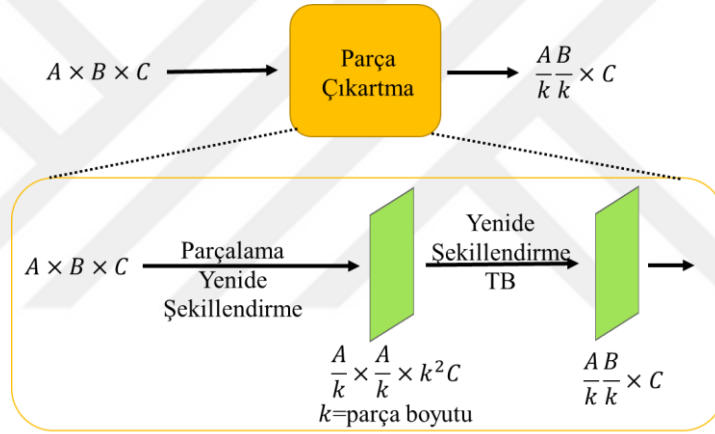
4.4.1.2 Melez dikkat kapısı

Yüzey hatasını, görüntüden ayırabilecek önemli özniteliklerin çıkarılması ve iyileştirilmesi/güçlendirilmesi, piksel seviyesinde yüzey hata tespiti için oldukça önemlidir. Özellikle renk, kenar, doku gibi mekânsal detaylar, küçük yüzey hatalarının tespitinde etkiliyken, küresel semantik öznitelikler arka plana benzer yüzey hatalarında etkilidir. Bu çalışmada hem küresel semantik hem de mekânsal detayları ortaya çıkarmak için Melez Dikkat Kapısı (MDK, Hybrid Attention Gate) geliştirilmiştir. Önerilen MDK modeli, Trans ve KSMU dikkat bloklarının etkili bir kombinasyonuna dayanmaktadır.



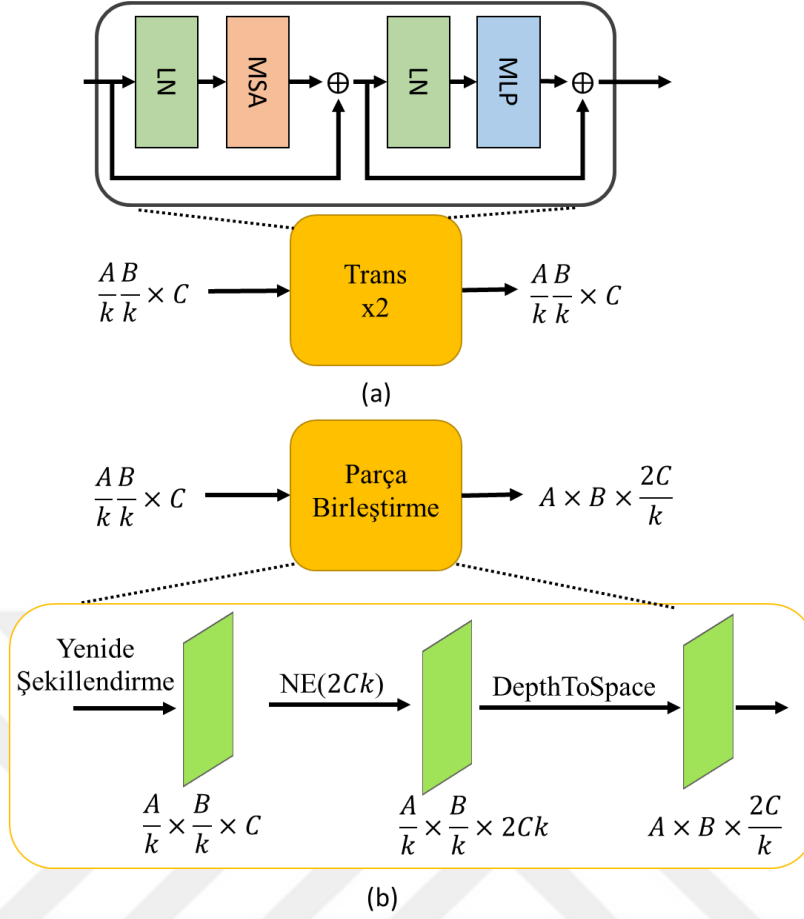
Şekil 4.22: Dönüştürücü (Trans) tabanlı dikkat Bloğu.

Dönüştürücü (Trans) tabanlı dikkat bloğu: Yüzey hata tespiti için önerilen Trans dikkat bloğu Şekil 4.22’de gösterilmiştir. Trans yapısının girişine verilen öznitelik haritasının boyutu $A \times B \times C$ (A genişlik, B yüksekliği ve C derinlik) olarak ele alınmıştır. Öncelikle, girdinin Trans bloğuna aktarılması için parçalama (tokenleştirme) işlemi gerçekleştirilmiştir. Bunun için girdi öznitelik haritası Parça Çıkartma (PÇ) bloğundan geçirilmiştir. PÇ’de girdi öznitelik haritası $k \times k \times C$ boyutunda parçalara bölünmüştür. Ardından, her bir parça vektörleştirildikten ($1 \times k^2 C$) sonra birleştirilmiştir. Bu işlem sonucunda $\frac{A}{k} \times \frac{B}{k} \times C$ boyutunda birleştirilmiş öznitelik vektörleri elde edilmiştir. PÇ’nin son kısmında elde edilen öznitelik vektörleri, tam bağlı (TB) katman uygulanarak $\frac{A}{k} \times \frac{B}{k} \times C$ boyutunda yeni öznitelik vektörleri elde edilmiştir. Yukarıdaki işlemler Şekil 4.23’ge gösterilmiştir.



Şekil 4.23: Parça Çıkartma Bloğu.

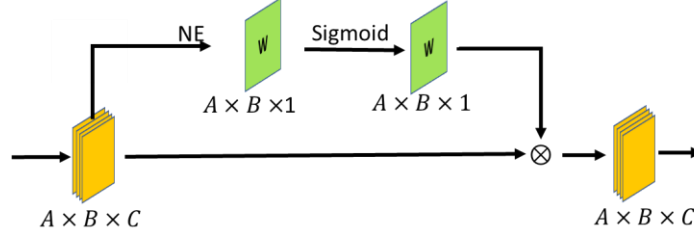
PÇ’nin çıkışında elde edilen vektörleştirilmiş öznitelikler dönüştürücü (Trans) bölümüne aktarılmaktadır. Dönüştürücü bölümünde Çok Kafalı Öz-Dikkat (MSA) ve Çok Katmanlı Projeksiyon (MLP) yapılarından oluşmaktadır. Şekil 4.24’te verildiği gibi, Trans bloğundaki her MSA veya MLP işleminden önce bir normalleştirme (LN) işlemi uygulanmıştır.



Şekil 4.24: MDK modelinin dönüştürücü mekanizması (a) ve parça birleştirme bloğu.

Dönüştürücü mekanizmasının çıkışı, $\frac{AB}{k^2} \times C$ boyutunda ve $\frac{AB}{k^2}$ adet parçanın her biri için bir vektörleştirilmiş öznitelik ($1 \times C$) içermektedir. Trans bloğunda bu öznitelikler sonraki katmanlarda kullanılmak üzere yeniden boyutlandırılmıştır. Yeniden boyutlandırma işlemleri için parça genişletme [118] bloğundan esinlenerek Parça Birleştirme (PB, Şekil 4.24.b) bloğu kullanılmıştır. PB bloğunda girdi öncelikle yeniden şekillendirilerek $\frac{A}{k} \times \frac{B}{k} \times C$ boyutuna getirilmiştir. Daha sonra $2Ck$ filtreye sahip bir Nokta Evrişim (NE) işlemi uygulanmıştır. İşlem sonucunda $\frac{A}{k} \times \frac{B}{k} \times 2Ck$ boyutunda öznitelik haritası elde edilmiştir. Son olarak, TensorFlow kitaplığında bulunan DepthToSpace (`tf.nn.depth_to_space`) katmanı uygulanarak bu öznitelik haritası $A \times B \times \frac{2C}{k}$ boyutuna yeniden şekillendirilmiştir. Sonuç olarak Şekil 4.22.b gösterilen Trans blok çıktısında $A \times B \times \frac{2C}{k}$ boyutunda bir öznitelik haritası elde edilmiştir.

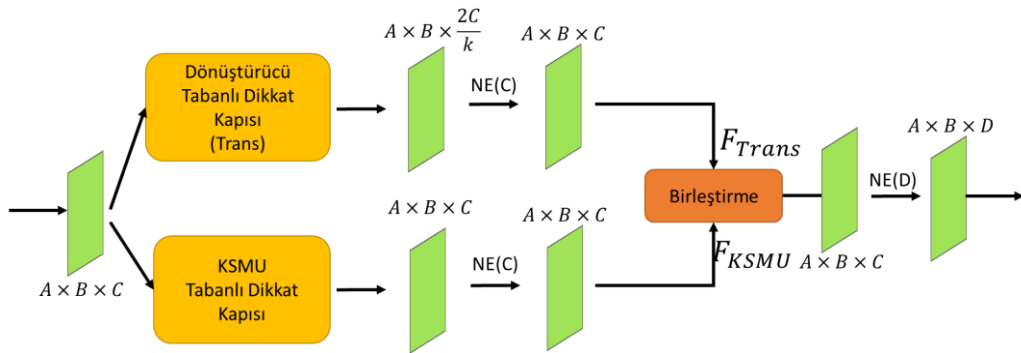
Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu: Bu çalışmada, Trans modelinde kaybolan mekânsal detayları güçlendirmek/çıkartmak için Kanal Sıkma ve Mekânsal Uyarlama (KSMU) yapısı kullanılmıştır [107].



Şekil 4.25: Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu.

Şekil 4.25'te görülebileceği gibi, KSMU bloğunun girdi öznetelik haritasına bir filtrelili noktasal evrişim uygulanmıştır. Noktasal evrişim sayesinde, öznetelik haritasındaki her noktanın derinlik boyutu bir noktada toplanmıştır. Daha sonra noktasal evrişim çıktısına sigmoid fonksiyonu uygulanarak matris ağırlıkları elde edilmiştir. Son olarak elde edilen ağırlıklar girdi ile eleman bazlı çarpma işlemi uygulanarak çıktı öznetelik haritası elde edilmiştir. Daha detaylı bilgi için Bölüm 2.3 bakılabilir.

KSMU ve Trans çıktılarını birleştirme: MDK modelinde KSMU ve Trans, girdi öznetelik haritasına paralel olarak uygulanmıştır. Trans'ta yer alan öz-dikkat mekanizması ile güçlü küresel anlamsal ayrıntıları çıkartılırken, KSMU ile önemli mekânsal (uzamsal) ayrıntıları çıkartılmaktadır. MDK modelinin son bölümünde KSMU ve Trans bloklarından elde edilen çıktılar birleştirilmiştir. Önerilen MDK birleştirme şeması Şekil 4.26'da verilmiştir.



Şekil 4.26: Melez Dikkat Kapısı (MDK, Hybrid Attention Gate).

KSMU ve Trans çıktıları birleştirmek için 4 farklı yaklaşım geliştirilmiştir. Bunlar doğrudan Birleştirme (*concat*), Toplama (*add*), Maksimum Çıkış (*add*) ve Çarpma (*mul*) yaklaşımlarıdır. Şekil 4.26'da görüleceği üzere, tüm birleştirme yaklaşımlarının

başında KSMU ve Trans çıktılarına noktasal evrişim (NE) işlemi uygulanmıştır. Daha sonra her birleştirme yaklaşımı için aşağıda detayları verilen işlemlerden geçirilmiştir. Aşağıdaki 4.7, 4.8, 4.9 ve 4.10 numaralı Denklemlerde F_{Trans} ve F_{KSMU} değerleri sırasıyla Trans ve KSMU çıktılarında elde edilen öznitelik haritasını temsil etmektedir.

Doğrudan Birleştirme (concat): Trans ve KSMU çıkışları doğrudan derinlik (kanal) boyutunda birleştirilmektedir. Doğrudan birleştirme modeli, veri kaybı olmadan iki farklı öznitelik haritasının kullanılmasına izin vermektedir. Ancak, bu hesaplama maliyetini arttırabilmektedir.

$$HAG_{A \times B \times 2C}^{concat}(i, j, n) = \text{Concat}(F_{SSE}(i, j, n), F_{Trans}(i, j, n)) \quad (4.7)$$

Toplama (add): Trans ve KSMU çıktıları, eleman bazında toplanmaktadır. Özniteliklerin toplanması, iki farklı Trans ve KSMU çıktılarına eşit öncelik vermektedir.

$$HAG_{A \times B \times C}^{add}(i, j, n) = F_{SSE}(i, j, n) \oplus F_{Trans}(i, j, n) \quad (4.8)$$

Maksimum Çıkış (Max-Out, max): Trans ve KSMU çıkışlarında büyük değer seçilmektedir. [107] nolu çalışmada belirtildiği gibi, maksimum çıktı, eleman bazında iki farklı çıktı arasındaki rekabeti zorlamaktadır. Bu durum, öznitelikler arasında seçici olmasını sağlamaktadır.

$$HAG_{A \times B \times C}^{max}(i, j, n) = \text{Max}(F_{SSE}(i, j, n), F_{Trans}(i, j, n)) \quad (4.9)$$

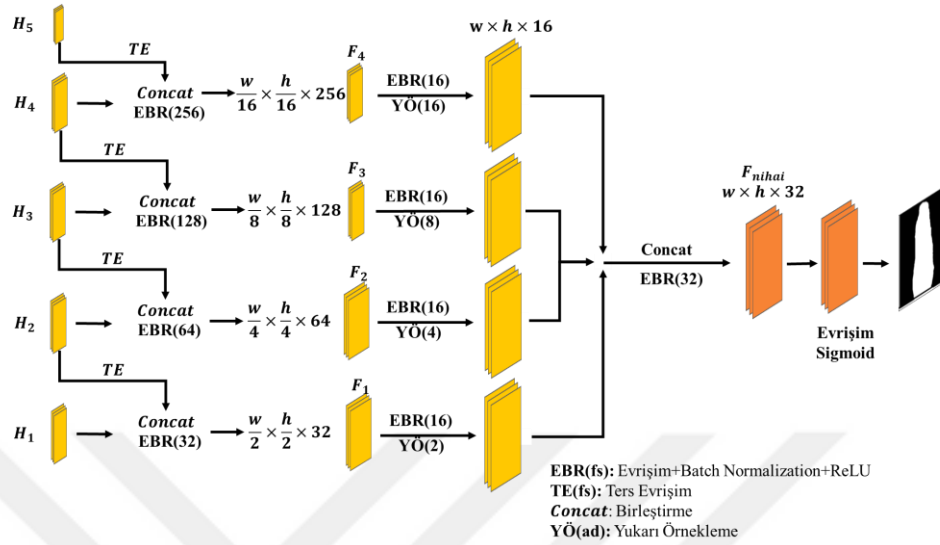
Çarpma (mul): Trans ve KSMU çıktılarına eleman bazında çarpma uygulanarak yeni bir öznitelik haritası oluşturulmaktadır. Burada önemli bir nokta KSMU ve Trans çıktılarında birbirinin aksine sonuçlar alındığında bazı önemli öznitelikler kaybolabilmektedir.

$$HAG_{A \times B \times C}^{mul}(i, j, n) = F_{SSE}(i, j, n) \odot F_{Trans}(i, j, n) \quad (4.10)$$

4.4.1.3 Öznitelik füzyon modülü

AlexNet, VGG ve Resnet gibi sınıflandırma mimarileri, son katmandaki öznitelik haritalarını kullanmaktadır. Bu mimarilerde uygulanan evrişim, ReLU ve havuzlama katmanları ile derin anlamsal öznitelik haritaları elde edilirken mekânsal detaylar kaybolabilmektedir [55, 85, 155]. Mekânsal bilgi taşıyan öznitelikler piksel seviyesinde hata tespitinde çok önemlidir. Bu bakış açısıyla geliştirilen birçok çalışmada, mekânsal bilgi

taşıyan düşük seviyeli öznitelikler, üst seviyeli öznitelikler ile birleştirilmiştir [7, 85]. Bu çalışmada, Önceki katmanlarda elde edilen çok seviyeli öznitelik haritalarını birleştirmek için öznitelik füzyon modülü geliştirilmiştir.



Şekil 4.27: Öznitelik Füzyon Modülü.

Şekil 4.27’de gösterildiği gibi, öznitelik füzyon modülünde, MDK modüllerinden elde edilen öznitelik haritaları hiyerarşik olarak birleştirilmiştir. İki farklı öznitelik haritasını birleştirmek için, yüksek seviyeli öznitelik haritası ($H_{yüksek\ seviye}$) ters evrişim (TE, ConvTranspose) katmanından geçirilmiştir. Bu sayede $H_{yüksek\ seviye}$ öznitelik haritası $H_{düşük\ seviye}$ öznitelik haritası ile aynı boyuta getirilmiştir. Ardından, iki öznitelik haritası kanal boyutunda birleştirilmiştir (Concat). Son olarak, birleştirilmiş öznitelik haritalarına sırası ile Evrişim, Batch normalizasyonu ve ReLU (EBR) katmanları uygulanmıştır. Bu süreç Denklem 4.11’deki gibi formülize edilmiştir.

$$F_{new} = \text{EBR} \left(\text{Concat} \left(\begin{array}{c} \text{TE}(H_{yüksek\ seviye}) \\ H_{düşük\ seviye} \end{array} \right) \right) \quad (4.11)$$

Burada EBR sırasıyla uygulanan evrişim, Batch normalleştirme ve ReLU katmanlarını temsil etmektedir.

Öznitelik füzyon modülünün ilk bölümünde, MDK modülünden beş farklı seviyede alınan öznitelik haritalarına, yukarıdaki işlemler uygulanarak dört farklı öznitelik haritası elde edilmiştir. Öznitelik füzyon modülünün son bölümünde, tüm öznitelik haritaları, yukarı örnekleme katmanı kullanılarak görüntü ile aynı boyutta getirilmiştir. Ardından, tüm

öznitelik haritaları, nihai öznitelik (F_{nihai}) haritası için kanal boyutunda birleştirilmiştir (Denklem 4.12).

$$F_{nihai} = Concat \begin{pmatrix} YÖ^{16}(F_1) \\ YÖ^8(F_2) \\ YÖ^4(F_3) \\ YÖ^2(F_4) \end{pmatrix} \quad (4.12)$$

Burada F_x öznitelik füzyon modülünün x' inci seviyede elde edilen öznitelik haritası ve $YÖ^{ad}$ adım değeri ad olan yukarı örnekleme katmanını temsil eder.

Şekil 4.27'de verildiği gibi öznitelik füzyon modülünün son katmanında elde edilen F_{nihai} öznitelik haritası $W \times H \times 64$ boyutundadır. Önerilen MDK-ÖFA ağındaki hata tespiti tahmin haritası için, son öznitelik haritasına bir evrişim (32 filtre), batch normalizasyon ve ReLU (EBR^{32}) katmanı uygulanmıştır. Daha sonra çekirdek boyutu 3x3 olan bir evrişim uygulanmıştır. Son olarak, sigmoid aktivasyon fonksiyonu (Denklem 4.13) uygulanarak tahmin haritası elde edilmiştir.

$$\text{Tahmin Haritası} = \sigma(\text{Evrişim}(EBR^{32}(F_{nihai}))) \quad (4.13)$$

Burada F_{nihai} , * ve σ sırasıyla nihai derin öznitelik haritalarını ve sigmoid aktivasyonunu temsil eder. Önerilen MDK-ÖFA mimarisinin katman yapıları ve bağlantıları Çizelge 4.18'de verilmiştir.

Çizelge 4.18: Önerilen MDK-ÖFA ağının yapısı (pb: parça boyutu, C ve D: şekil 4.26'deki NE'nin filtre sayısı).

Giriş	Modül	Detay	Çıktı
Görüntü	Öznelik çıkarma modülü	mobilenet v2	L2, L3, L4, L6, L8 öznelik haritası
L8 ($6^2 \times 1280$)	MDK 5. seviye	KSMU ve Trans pb:2, C=1280, D=512	Hibrit dikkat öznelik haritası (HA_5) $6^2 \times 1280$
L6 ($12^2 \times 576$)	MDK 4. seviye	KSMU ve Trans pb:2, C=576, D=256	HA_4 $12^2 \times 576$
L4 ($24^2 \times 192$)	MDK 3. seviye	KSMU ve Trans pb:4, C=192, D=128	HA_3 $24^2 \times 192$
L3 ($48^2 \times 144$)	MDK 2. seviye	KSMU ve Trans pb:4, , C=144, D=64	HA_2 $48^2 \times 144$
L2 ($96^2 \times 96$)	MDK 1. seviye	KSMU ve Trans pb: 8, C=96, D=32	HA_1 $96^2 \times 96$
MDK 5. ve 4. seviyeleri	Öznelik füzyon modülü 4. seviye		$192^2 \times 16$
MDK 4. ve 3. seviyeleri	Öznelik füzyon modülü 3. seviye		$192^2 \times 16$
MDK 3. ve 2. seviyeleri	Öznelik füzyon modülü 2. seviye	Şekil 4.27'deki ayrıntılar	$192^2 \times 16$
MDK 2. ve 1. seviyeleri	Öznelik füzyon modülü 1. seviye		$192^2 \times 16$
Öznelik füzyon modülünün 1.,2.,3. ve 4. seviyeleri	Öznelik füzyon modülünün çıkışı		$192^2 \times 32$
Öznelik füzyon modülünün çıkışı	Son katman	Evrişim ve sigmoid	$192^2 \times 1$

Önerilen MDK-ÖFA ağının eğitiminde İkili Çapraz Entropi (Binary Cross-Entropy), Denklem 4.3) kayıp fonksiyonu kullanılmıştır.

4.4.2 Deneysel çalışmalar ve sonuçları

Deneysel çalışmanın ilk bölümünde önerilen MDK modelinin performans sonuçları analiz edilmiştir. Bunun için MT ve MVTEC-Doku veri kümeleri kullanılarak MDK-ÖFA versiyonları (MDK^{concat}-ÖFA, MDK^{add}-ÖFA, MDK^{max}-ÖFA, MDK^{mul}-ÖFA, KSMU-ÖFA ve Trans-ÖFA) üzerinde bir takım deneysel çalışmalar yapılmıştır. Daha sonra performans sonuçları görsel ve sayısal olarak karşılaştırılmıştır. MDK-ÖFA modellerinin versiyonları Çizelge 4.19’da detaylandırılmıştır. Ek olarak, KSMU ve Trans bloklarının bireysel sonuçlarını analiz etmek için deneysel çalışmalara Trans-ÖFA ve KSMU-ÖFA modelleri eklenmiştir. Trans-ÖFA ve KSMU-ÖFA mimarilerinde sırasıyla atlama bağlantılarında sadece Trans ve KSMU blokları kullanılmıştır.

Çizelge 4.19: MDK-ÖFA ağlarının sürümleri ve bileşenleri.

Model Sürümleri	Bileşenler
MDK ^{concat} -ÖFA	MDK ^{concat} tabanlı önerilen model
MDK ^{add} -ÖFA	MDK ^{add} tabanlı önerilen model
MDK ^{max} -ÖFA	MDK ^{max} tabanlı önerilen model
MDK ^{mul} -ÖFA	MDK ^{mul} tabanlı önerilen model
KSMU-ÖFA	KSMU tabanlı önerilen model
Trans-ÖFA	Trans tabanlı önerilen model

Deneysel çalışmanın ikinci bölümünde önerilen model, Trans-UNet [116], Swin-UNet [118], Attention-UNet [104], FPN [66], LinkNet [64], PSPNet [134], UNet [67], Segnet [124] gibi son teknoloji modeller ile karşılaştırılmıştır

4.4.2.1 Veri kümeleri ve uygulama detayları

Bu çalışmada önerilen model Python platformu ile Tensorflow ve Keras kütüphaneleri kullanılarak geliştirilmiştir. Tüm deneylerde, I9 9900kf CPU, 64 GB RAM ve RTX 2080 ti GPU ve Ubuntu 20,04 sistemi kullanılmıştır. Önerilen modelin eğitim aşamasında kullanılan parametreler Çizelge 4.20’de verilmiştir.

Çizelge 4.20: Mevcut çalışmada kullanılan parametreler.

Parametreler	Değerler
Parti boyutu	8
Öğrenme oranı	0,0001
Eğitim Dönemi	100
Optimizasyon Yöntemi	Adam

Deneysel çalışmalarda, önerilen model performansını değerlendirmek için Precision, Recall, F1-sıkor ve mIoU metrikleri kullanılmıştır.

Deneysel çalışmalarda MT ve MV Tec-Doku veri kümeleri kullanılmıştır. Deneysel çalışmada, her bir veri kümesi eğitim için %50, doğrulama için %25 ve test için %25 olarak ayarlanmıştır. Ayrıca eğitim veri kümesindeki görüntülere döndürme (90° , 180° ve 270°), kırpma, bulanıklaştırma ve çevirme gibi veri arttırma yöntemlerini kullanarak veri kümesindeki örnek sayısı 8 katına çıkartılmıştır.

4.4.2.2 MDK birleştirme stratejilerinin sonuçları ve analizi

MT veri kümesinden elde edilen görsel sonuçlar Şekil 4.28’de verilmiştir. Şekil 4.28’de birinci ve ikinci sütunlar sırası ile görüntüyü ve gerçek değeri göstermektedir. Diğer sütunlar ise sırasıyla MDK^{concat}-ÖFA, MDK^{add}-ÖFA, MDK^{max}-ÖFA, MDK^{mul}-ÖFA, KSMU-ÖFA ve Trans-ÖFA için tahmin sonuçlarını göstermektedir. Şekil 4.28’de verilen 1. ve 2. sıradaki örneklerde küçük yüzey hataları içermektedir. Bu örneklerde Trans-ÖFA modeline göre KSMU-ÖFA modeli ile yüzey hataları daha başarılı bir şekilde tespit edilmiştir. Bu sonuçlarda, dönüştürücü modelinin küçük boyutlu yüzey hatalarının tespiti için zayıf olduğu görülmektedir. Bunun yanı sıra önerilen MDK^{concat}-ÖFA ağı ile bu küçük hatalar yüksek puanlar ile tespit edilmiştir. 3. sıradaki örneklerde yüzey hatası, arka plana benzerlik göstermektedir. Ayrıca bu örnekte ışıktan dolayı düşük kontrast sorunu vardır. Bu görüntüde MDK^{max}-ÖFA modeli kullanılarak yüzey hatası diğer tüm modellere göre daha iyi tespit edilmiştir. Bu görüntü için Trans-ÖFA ve KSMU-ÖFA modelleri ile yüzey hatası tam olarak tespit edilememiştir. 4. ve 5. sıralardaki yüzey hataları önceki örneklere göre daha belirgindir. Bu örneklerde, tüm modeller iyi performans sağlamıştır. Önerilen modeller arasında en iyi sonuç MDK^{max}-ÖFA ağı ile elde edilmiştir.

Row:1		iou:82.62	iou:73.78	iou:62.81	iou:63.79	iou:81.72	iou:30.81
Row:2		iou:47.09	iou:0.0	iou:33.83	iou:5.74	iou:36.45	iou:0.0
Row:3		iou:55.63	iou:84.86	iou:90.58	iou:78.97	iou:8.54	iou:43.43
Row:4		iou:78.09	iou:79.86	iou:80.64	iou:77.57	iou:78.18	iou:76.57
Row:5		iou:85.16	iou:86.42	iou:87.71	iou:86.45	iou:86.41	iou:88.62
Görüntü	Gerçek Değeri	MDK ^{concat} -ÖFA	MDK ^{add} -ÖFA	MDK ^{max} -ÖFA	MDK ^{mul} -ÖFA	KSMU-ÖFA	Trans-ÖFA

Şekil 4.28: MT veri kümesinden örnek görüntüler ve MDK-ÖFA modellerinin tahmin sonuçları.

MT veri kümesi için önerilen modellerin sayısal sonuçları Çizelge 4.21’de verilmiştir. Bu sonuçlara göre önerilen MDK-ÖFA modelleri arasında en yüksek puan MDK^{max}-ÖFA modeli (%83,83 mIoU) ile elde edilirken, ikinci en yüksek puan MDK^{add}-ÖFA modeli (%82,02 mIoU) ile elde edilmiştir. Ayrıca MDK^{concat}-ÖFA ve MDK^{mul}-ÖFA ağlarının mIoU puanları sırasıyla %83,83 ve %82,02 olarak elde edilmiştir. Sadece dönüştürücü bloklarına dayalı Trans-ÖFA ağı %79,10 mIoU skoruna ulaşırken, KSMU bloğuna dayalı KSMU-ÖFA ağı %81,38 mIoU skoruna ulaşmıştır.

Çizelge 4.21: MDK-ÖFA modellerinin MT veri kümesindeki performans sonuçları.

Modeller	mIoU	F1-skoru	Kesinlik	Duyarlılık
MDK ^{concat} -ÖFA	81,58	89,86	91,98	87,83
MDK ^{add} -ÖFA	82,02	90,12	93,45	87,02
MDK ^{max} -ÖFA	83,83	91,20	95,28	87,47
MDK ^{mul} -ÖFA	81,40	89,74	91,60	87,96
KSMU-ÖFA	81,38	89,73	95,40	84,69
Trans-ÖFA	79,10	88,33	93,79	83,47

MVTec doku veri kümesinde yüzey hatası değişen boyutlarda ve arka plan ile benzerlik görmektedir. Ayrıca kumaş ve mermer gibi 5 farklı yüzeyden çekilmiş görüntüleri içerdiği için arka plan oldukça değişkendir. MVTEC veri kümesinden elde edilen görsel sonuçlar Şekil 4.29’da verilmiştir. Şekil 4.29’da 3. ve 4. sıradaki yüzey hatalarının en yüksek

performansları MDK^{concat}-ÖFA ağ modeli ile elde edilirken, 1., 2., 5., 6. ve 9. sıralarda en yüksek performans MDK^{max}-ÖFA ağı ile elde edilmiştir. 7. ve 8. sıradaki deri yüzey görüntüsünde MDK^{add}-ÖFA modeli kullanılarak yüzey hatası diğer modellere göre daha iyi tespit edilmiştir.

Row:1		iou:74.64	iou:73.22	iou:77.03	iou:72.26	iou:73.14	iou:78.29
Row:2		iou:28.1	iou:46.71	iou:54.77	iou:32.07	iou:16.66	iou:28.74
Row:3		iou:71.01	iou:70.29	iou:68.99	iou:60.97	iou:67.41	iou:63.38
Row:4		iou:84.28	iou:82.56	iou:83.64	iou:44.93	iou:79.92	iou:78.36
Row:5		iou:77.74	iou:74.16	iou:78.09	iou:71.21	iou:79.06	iou:68.19
Row:6		iou:76.23	iou:64.74	iou:81.53	iou:74.4	iou:69.54	iou:80.39
Row:7		iou:70.12	iou:77.74	iou:72.92	iou:74.51	iou:71.65	iou:68.4
Row:8		iou:63.36	iou:70.87	iou:68.0	iou:59.8	iou:63.71	iou:66.96
Row:9		iou:24.39	iou:12.0	iou:39.66	iou:18.78	iou:0.0	iou:26.08
Görüntü	Gerçek Değeri	MDK ^{concat} -ÖFA	MDK ^{add} -ÖFA	MDK ^{max} -ÖFA	MDK ^{mul} -ÖFA	KSMU-ÖFA	Trans-ÖFA

Şekil 4.29: MVTec-Doku veri kümesi için önerilen ve diğer modellerin görsel tahmin sonuçları.

MVTec-Doku veri kümesinden elde edilen sayısal sonuçlar Çizelge 4.22’de verilmiştir. Bu sonuçlara göre en yüksek sonuç (%79,51 mIoU) MDK^{concat}-ÖFA ağı ile elde edilirken, ikinci en yüksek sonuç (%79,34 mIoU) MDK^{max}-ÖFA ile elde edilmiştir. Ek olarak, önerilen MDK^{add}-ÖFA ve MDK^{mul}-ÖFA ağları sırasıyla %77,36 ve %77,04 mIoU puanları üretmiştir. Bunun yanı sıra KSMU-ÖFA modeli kullanılarak ortalama bir skor (%78,55 mIoU) elde edilmiştir. En düşük puanlar ise (%75,85 mIoU) Trans-ÖFA ağı ile elde edilmiştir.

Çizelge 4.22: MDK-ÖFA modellerinin MVTEC-Doku veri kümesindeki performans sonuçları.

Modeller	mIoU	F1-skoru	Kesinlik	Duyarlılık
MDK ^{concat} -ÖFA	79,51	88,58	91,18	86,13
MDK ^{ekle} -ÖFA	77,36	87,23	85,42	89,12
MDK ^{max} -ÖFA	79,34	88,48	89,43	87,55
MDK ^{mul} -ÖFA	77,04	87,03	87,77	86,30
KSMU-ÖFA	78,55	87,99	89,11	86,89
Trans-ÖFA	75,85	86,26	85,46	87,08

Sonuç olarak MT’de ele alındığı gibi, Trans modeli küresel anlamsal ayrıntıları yakalamada oldukça iyi olmasına rağmen küçük mekânsal ayrıntıları yakalamakta zayıftır. MT veri kümesinin aksine, MVTEC-doku veri kümesi çok daha küçük yüzey hatalarına sahiptir. Ayrıca arka planda düşük kontrast sorunu yoktur. Bu nedenle, tek başına Trans modeli ile en düşük sonuçlar elde edilirken, KSMU ve Trans modelini birlikte kullanan MDK^{max}-ÖFA ve MDK^{concat}-ÖFA modelleri ile en yüksek sonuçlar elde edilmiştir. Ayrıca Trans öznelik çıkışları KSMU öznelik çıkışlarını bastırıldığından, KSMU-ÖFA modeline göre, MDK^{add}-ÖFA ve MDK^{mul}-ÖFA modelleri ile daha düşük sonuçlar elde edilmiştir.

4.4.2.3 Son teknoloji modeller ile karşılaştırma

Önerilen modelin farklı versiyonlarının sayısal ve görsel analizleri önceki bölümde detaylandırılmıştır. Bu bölümde aynı deneysel çalışmalar Trans-UNet, Swin-UNet, Attention-UNet, FPNet, LinkNet, PSPNet, UNet gibi modellere uygulanmıştır. Bu bölümde her iki veri kümesinde de yüksek sonuçlara sahip olan MDK^{max}-ÖFA modelinin sonuçları kullanılmıştır.

MDK^{max}-ÖFA ve diğer modellerden elde edilen performans sonuçları Çizelge 4.23’te verilmiştir. Attention-UNet ağı gelişmiş bir model olmasına rağmen, Attention-UNet ile düşük sonuçlar (%66,26 mIoU) elde edilmiştir. Bunun sebebinin Attention-UNet ağının, MT veri kümesinde sıklıkla karşılaşılan düşük kontrast problemi karşısında zayıf kalmasıdır. [116] nolu çalışmada ve önceki bölümdeki analizlerde görüldüğü gibi, dönüştürücü bloklarının küçük mekânsal detayları yakalamada zayıf olduğundan en düşük sonuçlar Swin-UNet ağı ile elde edilmiştir. Bunun yanı sıra tüm katmanlarda dönüştürücü kullanılan Swin-UNet ağının aksine, sadece son katmanında dönüştürücü kullanan Trans-UNet ağı %72,18 mIoU performansına ulaşmıştır. Öte yandan, dönüştürücü ve KSMU bloklarını ortak bir şekilde kullanan MDK^{max}-ÖFA modeli, Trans-UNet modeline göre %83,83 mIoU skoru ile daha yüksek performansa ulaşmıştır.

Çizelge 4.23: MT veri kümesi için MDK^{max}-ÖFA ve son teknoloji modellerin performans sonuçları.

Modeller	mIoU	F1-skoru	Kesinlik	Duyarlılık
MDK ^{max} -ÖFA	83,83	91,20	95,28	87,47
Trans-UNet	72,18	83,84	89,23	79,06
Swin-UNet	12,39	22,06	51,80	14,01
Attention-UNet	66,26	79,70	87,36	73,28
FPNet	73,98	85,05	90,85	79,94
LinkNet	69,80	82,21	91,90	74,37
PSPNet	60,42	75,33	80,37	70,88
UNet	70,63	82,78	91,25	75,75
SegNet	59,50	74,61	90,89	63,27

MVTec veri kümesine dayalı diğer modellerin performans sonuçları Çizelge 4.24'te verilmiştir. Bu sonuçlara göre önerilen MDK^{max}-ÖFA ağı ile en yüksek başarı (%79,34 mIoU) elde edilirken, Trans-UNet ile ikinci en yüksek başarı (74,48 mIoU) elde edilmiştir. Attention-UNet, FPNet, UNet, LinkNet, PSPNet ve SegNet ağları sırasıyla %72,17, %73,47, %73,93, %67,84, %65,40 ve %58,34 mIoU puanlarına ulaşmıştır. %54,04 mIoU ile en düşük sonuçlar MT veri kümesinde olduğu gibi Swin-UNet ağı ile elde edilmiştir.

Çizelge 4.24: MVTEC-Doku veri kümesi için MDK^{max}-ÖFA ve son teknoloji modellerin performans sonuçları.

Modeller	mIoU	F1-skoru	Kesinlik	Duyarlılık
MDK ^{max} -ÖFA	79,34	88,48	89,43	87,55
Trans-UNet	74,48	85,37	88,98	82,05
Swin-UNet	54,04	70,16	87,71	58,47
Attention-UNet	72,17	83,83	86,44	81,38
FPNet	73,47	84,70	85,06	84,35
LinkNet	67,84	80,84	81,28	80,41
PSPNet	65,40	79,08	84,94	73,97
UNet	73,93	85,01	83,76	86,30
SegNet	58,34	73,69	90,97	61,93

4.5 Uygulama 5: Swin Dönüştürücü Tabanlı Çok Özellikli Entegrasyon Ağ Mimarisi

Otomatik yüzey hata tespiti için geliştirilen güncel yöntemlerde, yüksek başarılar sağlayan öznitelikleri güçlendirmek için Dikkat Kapısı ve Sıkma-Uyarlama yapıları kullanılmıştır [7, 41, 83, 105, 107]. Ağırlıklı olarak doğal dil işleme için geliştirilen dönüştürücüler, ViT modeli ile birçok bilgisayarlı görü alanında uygulanmış ve başarısı ispatlanmıştır [103]. Ancak [111] ve [118] nolu çalışmalarda belirtildiği gibi görüntü ve kelime yapıları farklı olduğu için dönüştürücü yapısı bölütleme ve algılama problemlerinde yeterince etkili olamamıştır. Bu probleme dayanarak, Liu ve dig. [111] segmentasyon ve algılama için Swin Dönüştürücü (Swin Transformer) yapısını geliştirmişlerdir. Swin dönüştürücüsü yerel pencere modeli ile ViT modelinden daha etkili sonuçlar sağlamıştır [111].

Swin-Dönüştürücüden ilham alan Cao ve dig. [118], Swin dönüştürücü yapısına dayalı Swin-UNet mimarisi geliştirmişlerdir. Bu model büyük veri kümelerinde test edilmiş ve etkili olduğu kanıtlanmıştır. ESA tabanlı yöntemlere göre daha üstün performans gösteren dönüştürücüler büyük veri kümelerine ihtiyaç duymuştur [118, 156]. Bununla birlikte yüzey hata tespiti için kullanılan veri kümeleri genellikle düşük sayıda örneklerle sahiptir. Bu kapsamda yapılan deneysel çalışmalarımızda Swin-UNet modeli yüzey hata tespiti için düşük sonuçlar üretmiştir.

Yukarıdaki problemde yola çıkarak, bu uygulamada piksel seviyesinde yüzey hatalarının tespiti için "önceden eğitilmiş Inception mimarisi" ve "Swin dönüştürücü tabanlı kod çözücü" ye dayalı yeni bir ağ mimarisi önerilmiştir. Swin dönüştürücü tabanlı Çok Özellikli Entegrasyon Ağı (Swin-ÇÖEA, Swin transformer-based Multi-Feature Integration Network) olarak adlandırılan bu ağ mimarisi, Swin-UNet mimarisinden farklı olarak kodlayıcı için Inception omurga ağı kullanmaktadır. Böylece dönüştürücüler için güçlü başlangıç öznitelikleri elde edilmiştir. Kod çözücü bölümünde, günümüzün en yeni dönüştürücü teknolojisi olan Swin dönüştürücü bloğu kullanılarak başlangıç özniteliklerinden küresel anlamsal öznitelikler (uzun menzilli bağlamsal bilgisi, long-range contextual information) elde edilmiştir. Bunun yanı sıra, önemli kenar, renk ve doku gibi küçük mekânsal (uzamsal) ayrıntıları elde etmede dönüştürücüler zayıf olduğundan, kod çözücünün son seviyesinde evrişim katmanı kullanılmıştır. Önerilen modelin Çok Özellikli Entegrasyon (ÇÖE) olarak adlandırılan son bölümünde, kod çözücünün farklı seviyelerinden

alınan öznitelik haritaları birleştirilmiştir. Daha sonra önemli öznitelikleri ortaya çıkarmak için Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu uygulanmıştır. Son olarak, evrişim ve sigmoid işlevi uygulanarak tahmin haritası elde edilmiştir. Önerilen modelin performansı MT ve MV Tec-Doku veri kümeleri ile analiz edilmiştir.

4.5.1 Önerilen model

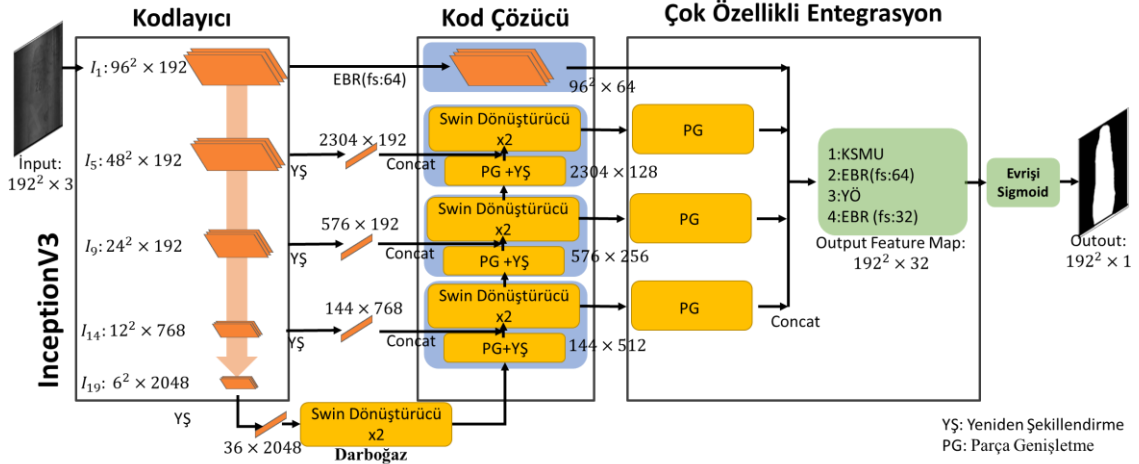
Önerilen model, Şekil 4.30'da gösterildiği gibi üç aşamadan oluşmaktadır: Kodlayıcı, Swin dönüştürücüsü tabanlı kod çözücü ve Çok Özellikli Entegrasyon (ÇÖE). Bu aşamaların başlıca özellikleri aşağıdaki gibidir.

a) Önerilen ağın kodlayıcı kısmında IncepitonV3 omurga ağı kullanılarak farklı seviyelerde başlangıç öznitelik haritaları elde edilmiştir. Ayrıca, Şekil 4.30'da görüldüğü gibi, darboğaz (bottleneck) bloğunda, kodlayıcı bölümünün son katman çıkışına Swin dönüştürücüsü uygulanmıştır.

b) Kod çözücü bölümü, kodlayıcı bölümünde olduğu gibi dört seviyeden oluşmaktadır. İlk üç seviyede Swin Dönüştürücü blokları kullanılırken son seviyede evrişim katmanı kullanılmıştır. Kod çözücüde hata tespiti için Swin dönüştürücü ile küresel semantik öznitelik haritaları çıkartılırken, evrişim katmanı ile kenar, renk ve doku bilgileri içeren mekânsal öznitelik haritaları çıkartılmıştır.

c) ÇÖE modülünde, kod çözücü bölümünden elde edilen dört farklı öznitelik haritası, parça genişletme katmanı ile boyutları büyütüldükten sonra birleştirilmiştir. Birleştirilmiş öznitelik haritası, hata tespiti için önemli mekânsal ve anlamsal ayrıntılar içermektedir. Son olarak önemli öznitelikleri ortaya çıkarmak için bu öznitelik haritalarına Kanal Sıkma ve Mekânsal Uyarlama (KSMU) bloğu uygulanmıştır.

d) Swin-ÇÖEA ağının son bölümünde, ÇÖE çıkışındaki özniteliklere evrişim ve sigmoid aktivasyon katmanı uygulanmış ve piksel seviyesinde yüzey hata tespit çıktısı elde edilmiştir.



Şekil 4.30: Önerilen Swin-ÇÖEA mimarisi.

4.5.1.1 Kodlayıcı

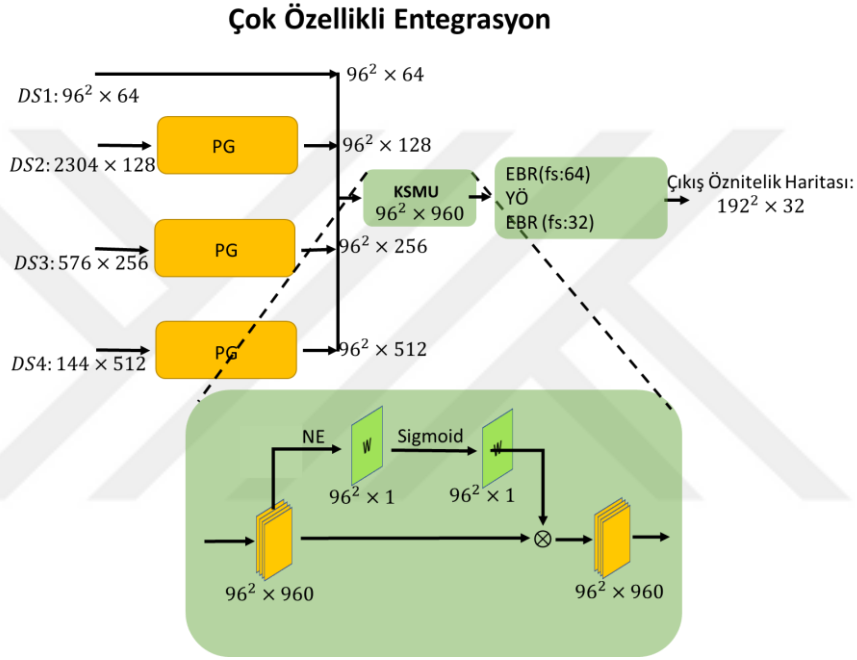
Piksel seviyesinde yüzey hata tespiti için mevcut veri kümeleri sınırlı sayıda örneğe sahiptir. Bu nedenle, doğrudan dönüştürücü tabanlı bir segmentasyon ağı kullanmak düşük sonuçlar vermiştir. Önerilen modelde, bu sorunla başa çıkmak için önceden eğitilmiş InceptionV3 ağı kullanılmıştır. InceptionV3 mimarisinin detayları Çizelge 4.1’de verilmiştir. InceptionV3 ağının I_1 , I_5 , I_9 , ve I_{14} katmanları kod çözücü bölümüne aktarılırken, I_{19} katman çıkışı darboğaz bloğuna aktarılmıştır.

4.5.1.2 Kod çözücü

Önerilen modelde Swin dönüştürücüler ve geleneksel evrişim katmanı kullanarak daha etkili bir hibrit kod çözücü modeli önerilmiştir. Bu kod çözücü dört aşamadan oluşmaktadır. Şekil 4.30’da görüldüğü gibi kodlayıcı bölümünün çıkışına iki adet Swin dönüştürücü (darboğaz ünitesi) uygulanmıştır. Daha sonra darboğaz çıkışı Kod çözücü bölümünün girişine aktarılmıştır. Kod çözücü bölümünün ilk aşamasında, Parça Genişletme (PG, bölüm 2.3.4’te detaylandırılmıştır) ile girdi öznelik haritasının boyutu artırılmıştır. Daha sonra, bu öznelik haritası, kodlayıcı bölümünün aynı seviyesinde çıktı öznelik haritası ile birleştirilmiştir (concat). Son olarak birleştirilen öznelik haritalarına Swin dönüştürücü bloğu uygulanmıştır. Bu işlem, kod çözücü bölümünün ilk üç aşamasında benzer şekilde tekrarlanmıştır. [116, 117] nolu çalışmalarda belirtildiği gibi dönüştürücü yapıları kenar, renk ve doku gibi mekânsal ayrıntıları ortaya çıkarmada zayıf kaldığı için kod çözücünün son aşamasında Swin dönüştürücü kullanmak yerine evrişim katmanı yapısı kullanılmıştır.

4.5.1.3 Çok özellikli entegrasyonu

UNet tabanlı mimariler, kod çözücü bölümünün son katmanındaki öznetelik haritasını kullanırken, FPNet tabanlı modellerde, kod çözücünün tüm seviyelerdeki öznetelik haritalarını birleştirerek kullanmıştır. FPNet modeli, piksel seviyesinde hata tespiti için orta katmandaki öznetelikleri doğrudan son katmana taşıyarak [7, 66] UNet mimarisine göre daha etkili bir yapı oluşturmuştur. Swin-ÇÖEA'in Çok Özellikli Entegrasyon (ÇÖE) modülü, FPNet modelinden esinlenerek geliştirilmiştir. ÇÖE modülünde temel olarak farklı kod çözücü aşamalarındaki öznetelikler tek bir öznetelik haritasında birleştirilmiştir (Şekil 4.31).



Şekil 4.31: Çok özellikli entegrasyon modülü.

Şekil 4.31’de gösterildiği gibi, ÇÖE modülü Swin-ÇÖEA ağının kod çözücüsünde dört farklı öznetelik haritası (DS1, DS2, DS3, DS4) ile beslenmektedir. Bu öznetelik haritaları sırasıyla $96^2 \times 64$, 576×256 , 144×512 , ve 2304×128 boyutlarındadır. ÇÖE modülünde, PG bloğu kullanılarak öznetelik haritalarının spektral boyutları 96×96 (genişlik \times yükseklik) değerine yükseltilmiştir. Daha sonra bu öznetelik haritaları, derinlik boyutunda *concat* katmanı ile birleştirilmiştir. Birleştirme işleminden sonra elde edilen F_{concat} öznetelik haritasına bir EBR katmanı (Evrişim, BN, ReLU) uygulanmıştır.

F_{concat} öznetelik haritası, kod çözücü modülünün farklı düzeylerinden beslendiği için çok seviyeli anlamsal ve mekânsal ayrıntıları içermektedir. Bu detaylardan önemli detayları çıkarmak için Kanal Sıkma ve Mekânsal Uyarılama Bloğu (KSMU) [107] uygulanmıştır.

Sonuç olarak, elde edilen F_{KSMU} öznetelik haritasında hata tespiti için önemli öznetelikler güçlendirilmiştir. Yukarıdaki işlemler Denklem 4.14'te formülize edilmiştir.

$$F_{concat} = CBNRelu^{fs=32} \left(Concat(F_1, PG^2(F_2), PG^4(F_3), PG^8(F_4)) \right) \quad (4.14)$$

$$F_{KSMU} = KSMU(F_{concat})$$

Denklem 4.14'te kullanılan ifadeler şunları temsil etmektedir:

- F_{concat} : birleştirilmiş öznetelik haritası
- F_i : Kod çözücünün i 'inci öznetelik haritasının
- \odot : eleman bazında çarpma
- $PG^{ad}(\cdot)$: ad genişleme oranına sahip parça genişletme katmanı

ÇÖE'nin sonunda nihai öznetelik (F_{nihai}) haritasını elde etmek için, F_{KSMU} öznetelik haritasına sırasıyla EBR, Yukarı Örneklemeye (YÖ) ve EBR katmanları uygulanmıştır. Son olarak hata tespiti tahmin haritasını elde etmek için evrişim ve sigmoid katmanı kullanılmıştır. Yukarıdaki işlemler Denklem 4.15'te formülize edilmiştir.

$$F_{nihai} = EBR_{3 \times 3}^{f=32} \left(YÖ^{s=2} \left(EBR_{3 \times 3}^{f=64} (F_{SE}) \right) \right) \quad (4.15)$$

$$P = \sigma(Evrişim_{3 \times 3}^1(F_{nihai}))$$

Denklem 4.15'te kullanılan ifadeler şunları temsil etmektedir:

- σ : Sigmoid aktivasyon fonksiyonu
- $E_{3 \times 3}^1$: sırasıyla çekirdek boyutu 3x3 ve filtre sayısı=1 olan Evrişim katman
- P : hata tespiti tahmin haritası
- $YÖ(\cdot)$: Boyut yükseltme için kullanılan yukarı örneklemeye katmanını

4.5.1.4 Modelin genel görünümü ve eğitimi

Önerilen Swin-ÇÖEA modeli, önceden eğitilmiş InceptionV3 ağı, Swin dönüştürücü, PG ve KSMU bloklarından oluşan hibrit bir modeldir. Swin-ÇÖEA mimarisinin katmanları ve katman parametreleri Çizelge 4.25'te gösterilmiştir.

Çizelge 4.25: Swin-ÇÖEA ağının katman ve parametre ayarları. (PS: dönüştürücü için girdinin parça sayısı; Head: dönüştürücüdeki kullanılan dikkat başlık sayısı; mlp : MLP’de için nöron sayısı).

Bağlantı	Modül		Detay
Görüntü	Kodlayıcı	InceptionV3	Detaylar Çizelge 4.1’de
I_{19}	Dar boğaz	Swin-Dönüştürücü x2	PS:36, Head:8, Pencere boyutu:4, mlp: 512, gömme :1024
		PG	
I_{14}, I_{19}	DS4	Swin-Dönüştürücü x2	PS:36, Head:8, Pencere boyutu:4, mlp:512, gömme:512
		PG	
$I_9, DS4$	DS3	Swin-Dönüştürücü x2	PS:144, Head:8, Pencere boyutu: 16, mlp : 512, embeddim:256
		PG	
$I_5, DS3$	DS2	Swin-Dönüştürücü x2	PS:576, Head:4, Pencere boyutu:16 mlp : 512, embeddim:128
		PG	
$I_1, DS2$	DS1	EBR	fs:64, çb:3x 3,ad:1
DS 1, DS 2, DS 3, DS 4,	ÇÖE	PG ve Concat	ad(stride):DS 3, DS 2, DS 1 için sırasıyla 2, 4 ve 8.
		KSMU	fs:1, çb: 1x1
		EBR	fs:64, çb:3x 3,ad :1
		YÖ	ad: 2
		EBR	fs:32, çb:3x 3,ad :1
ÇÖE	Swin-ÇÖEA çıkışı	Evrişim.	fs:1, çb:3x 3,ad :1
		Sigmoid	

Swin-ÇÖEA ağ mimarisinde omurga ağının başlangıç parametreleri için ImageNet üzerinde önceden eğitilmiş InceptionV3 ağının parametreleri kullanılmıştır. Swin-ÇÖEA mimarisinin diğer katmanlarına rastgele değerler atanmıştır. Eğitim boyunca Swin-ÇÖEA mimarisinin tüm parametreleri transfer öğrenme modeli ile güncellenmiştir. Swin-ÇÖEA ağ mimarisinin eğitiminde İkili Çapraz Entropi (Binary Cross-Entropy, Denklem 4.3) kayıp fonksiyonu kullanılmıştır.

4.5.2 Deneysel çalışmalar ve sonuçları

Deneysel çalışmalarda MT ve MV Tec-Doku veri kümeleri kullanılarak önerilen modelin performans analizi yapılmıştır. Ayrıca önerilen model LinkNet [64], PSPNet [134], FPNNet

[66], U-net [67] Attention-UNet [104] ve Swin-UNet [118] gibi son teknoloji modeller ile performans kıyaslamaları yapılmıştır.

4.5.2.1 Veri kümeleri ve uygulama detayları

Swin-ÇÖEA ve diğer son teknoloji modellerin performansını karşılaştırmak için MT ve MVTec-Doku veri kümeleri kullanılmıştır. Bu veri kümesinin detayları Bölüm 3'te verilmiştir. Deneysel çalışmalarda 2 kat çapraz doğrulama yöntemi kullanılmıştır. Bu doğrultuda Fold1 ve Fold2'nin örnek sayıları Çizelge 4.26'da verilmiştir.

Çizelge 4.26: k=2 çapraz doğrulama için, Fold 1 ve Fold 2'de bulunan hatalı ve hatasız görüntülerin sayısı.

Veri Kümesi	Fold 1	Fold 2
	Hatalı / Hatasız / Toplam	Hatalı / Hatasız / Toplam
MT	193 / 479 / 672	193 / 479 / 672
MVTec-Doku	191 / 700 / 891	191 / 700 / 891

Tüm deneyler, i9 9900kf CPU ve RTX 2080'ti GPU kullanılarak Ubuntu 20,04 sistemi altında yapılmıştır. Deneysel çalışmalarda modellerinin eğitim parametreleri aşağıdaki gibi ayarlanmıştır.

- Epoch:100
- Batch boyutu:8
- Öğrenme oranı: 0,001
- Optimizasyon yöntemi: Adam

Deneysel çalışmalarda, önerilen model performansını değerlendirmek için mIoU ve F1-skor metrikleri kullanılmıştır.

4.5.2.2 MT veri kümesi

MT veri kümesinden alınan örnek görüntü, temel gerçekleri ve modellerin tahmin çıktıkları Şekil 4.32'de gösterilmiştir. Şekil 4.32'de görüldüğü gibi önerilen Swin-ÇÖEA ağı diğer yöntemlerden daha iyi sonuç vermiştir. Şekil 4.32'de verilen 1. sıradaki örnek, arka plana oldukça benzeyen bir yüzey hatası içermektedir. Bu görüntüde Swin-ÇÖEA ağı kullanılarak diğer yöntemlere göre daha yüksek doğrulukla hata tespiti yapılmıştır. Ayrıca bu örnekte LinkNet ve UNet ağları yüzey hatasını kabaca tespit ederken, FPNet, PSPNet, Attention-UNet ve U-Net yöntemleri daha düşük puanlar üretmiştir. Öte yandan Swin-UNet modeli yüzey hatasını tespit edememiştir. 2. sıradaki görüntüde yüzey hataları belirgin olmasına rağmen, Swin-UNet, Attention-UNet ve LinkNet ağları, ile yüzey hatasını tam olarak tespit

edilememiştir. 2. sıradaki görüntüde yüzey hataları belirgin olmasına rağmen, Swin-UNet, Attention-UNet ve LinkNet ağları ile yüzey hatası tam olarak tespit edilememiştir. Buna kıyasla Swin-ÇÖEA ağı ile en yüksek doğrulukla hata tespiti yapılmıştır. 3. ve 4. sıradaki örneklerde küçük boyutlu yüzey hataları içermektedir. Bunun yanı sıra bu örneklerde değişen ışık problemi vardır. Bu zorluklardan dolayı yöntemler düşük sonuçlar üretmiştir. Ancak Swin-ÇÖEA ağı bu problemler karşısında daha iyi sonuçlar sağlamıştır. 5. sırada verilen görselde hatanın kenarları belirginken orta bölge arka plana benzerlik gösterdiği için tespit edilmesi zor bir yüzey hatasıdır. Bu nedenle, Attention-UNet, Swin-UNet, FPNNet ve PSPNet modelleri ile hatanın kenarları tespit edilirken orta bölge hatasız olarak tahmin edilmiştir. Bu sonuçlara kıyasla UNet ve Swin-ÇÖEA mimarileri ile daha başarılı sonuçlar elde edilmiştir.

1		GT	iou:47.3	iou:94.3	iou:52.52	iou:85.06	iou:75.23	iou:0.0	iou:96.66
2		GT	iou:58.39	iou:65.38	iou:87.34	iou:86.98	iou:81.6	iou:13.63	iou:88.34
3		GT	iou:53.12	iou:18.75	iou:0.0	iou:43.75	iou:56.81	iou:0.0	iou:85.71
4		GT	iou:33.33	iou:21.05	iou:0.0	iou:33.33	iou:28.88	iou:0.0	iou:74.35
5		GT	iou:31.83	iou:50.86	iou:59.58	iou:67.48	iou:2.13	iou:12.99	iou:79.18
	Image	Gerçek Değeri	FPNet	LinkNet	PSPNet	UNet	Attention-UNet	Swin-Unet	Swin-ÇÖEA

Şekil 4.32: MT görsel sonuçları, GT: Gerçek Değer.

Çizelge 4.27’de, MT’nin Fold1, Fold2 ve ortalama sonuçları verilmiştir. Modellerin performans değerlendirmeleri ortalama sonuçlarına göre yapılmıştır. Bu sonuçlara göre en yüksek sonuçlar (%81,37 mIoU ve %89,73 F1 skoru) Swin-ÇÖEA mimarisi ile elde edilmiştir. İkinci sırada FPNNet ağı ile %73,94 mIoU ve %85,01 F1-skoru elde edilmiştir. En düşük sonuçlar Swin dönüştürücü tabanlı Swin-Unet mimarisi ile elde edilmiştir. Swin-ÇÖEA ve Swin-UNet mimarileri karşılaştırıldığında önerilen modelin etkili ve başarılı sonuçlar verdiği kanıtlanmıştır.

Çizelge 4.27: MT veri kümesinin test sonuçları.

Model	Ortalama Puanlar		Fold1		Fold2	
	mIoU	F1-Skoru	mIoU	F1-Skoru	mIoU	F1-Skoru
Swin-ÇÖEA	81,37	89,73	81,11	89,57	81,63	89,89
Swin-UNet	17,28	29,46	16,81	28,78	17,74	30,14
Attention-UNet	55,39	70,98	47,76	64,65	63,01	77,30
FPNet	73,94	85,01	75,52	86,05	72,36	83,96
LinkNet	66,33	79,55	59,49	74,60	73,16	84,50
PSPNet	73,41	84,65	71,41	83,32	75,40	85,97
U-Net	72,31	83,93	7185	8362	7277	8424

4.5.2.3 MVTec-doku

MVTec-Doku veri kümesinden alınan örnek görüntüler ve modellerin tahmin sonuçları Şekil 4.33'te verilmiştir. Şekil 4.33'te verilen 1. ve 2. sıradaki örnekler incelendiğinde, Attention-UNet, FPNet ve LinkNet modelleri ile yüzey hataları kısmen tespit edilmişken, Swin-UNet, PSPNet ve UNet ağları ile yüzey hatası tespit edilememiştir. Öte yandan bu modellere kıyasla Swin-ÇÖEA modeli ile bu örneklerdeki hatalar daha yüksek performansla tespit edilmiştir. 3. sırada gösterilen ahşap yüzey örneği, arka plan ile uyumlu ve tespit edilmesi zor olan üç yüzey hatası içermektedir. Attention-UNet, FPNet, LinkNet, Swin-UNet, PSPNet ve UNet ağları ile bu yüzey hatalarından sadece biri tespit edilmişken, Swin-ÇÖEA modeli ile tüm yüzey hataları tespit edilmiştir. 4. sırada yer alan hatanın üst kısmı belirgin olduğu için tüm yöntemler bu bölgeyi tespit etmede başarılı sonuçlar sağlamıştır. Fakat alt bölgelerde bulunan hata parçalarını, arka plandan ayırmak zor olduğu için Swin-ÇÖEA hariç diğer yöntemler ile tespit edilememiştir. Sonuç olarak Şekil 4.33'te verilen örnek görüntülerde Swin-ÇÖEA modeli kullandığı InceptionV3 omurga ağı ve güçlü kod çözücü mimarisi ile diğer modellere göre üstün performans sağladığı gözlemlenmiştir.

1									
		GT	IoU:23.92	IoU:43.33	IoU:1.58	IoU:7.7	IoU:28.86	IoU:0.0	IoU:64.52
2									
		GT	IoU:32.5	IoU:5.34	IoU:22.0	IoU:45.2	IoU:0.0	IoU:0.0	IoU:58.26
3									
		GT	IoU:9.07	IoU:11.18	IoU:13.72	IoU:9.31	IoU:24.21	IoU:3.13	IoU:58.71
4									
		GT	IoU:42.23	IoU:37.55	IoU:26.29	IoU:43.04	IoU:24.59	IoU:31.75	IoU:57.14
5									
		GT	IoU:40.66	IoU:16.76	IoU:0.0	IoU:38.6	IoU:9.43	IoU:0.0	IoU:58.99
	Image	Gerçek Değeri	FPNet	LinkNet	PSPNet	UNet	Attention-Unet	Swin-Unet	Swin-ÇÖEA

Şekil 4.33: MVTec-Doku görsel sonuçları, GT: Gerçek Değer.

Çizelge 4.28, MVTec veri kümesinin sayısal sonuçlarını göstermektedir. Değerlendirmeler, MT veri kümesinde olduğu gibi ortalama sonuçlara göre yapılmıştır. Çizelge 4.28’de verilen sonuçlara göre Swin-ÇÖEA ağı ile en yüksek skorlar (%77,07 mIoU ve %87,05 F1 skor) elde edilirken, Swin-UNet ile en düşük sonuçlar (%51,62 mIoU ve %29,46 F1 skor) elde edilmiştir. Swin dönüştürücü kullanan Swin-Unet ve önerilen Swin-ÇÖEA mimarilerinin sonuçları kıyaslandığında, önerilen modelde swin dönüştürücüleri başarılı bir şekilde kullanıldığı ispatlanmıştır. Çizelge 4.28’de verilen ve geleneksel evrişim katmanı ile geliştirilen FPNet, UNet, LinkNet ve PSPNet ağları kullanılarak sırasıyla %74,51, %71,11, %65,21 ve %64,93 mIoU skorları elde edilmiştir. Bu modeller arasında FPNet ağı ile en yüksek sonuçlar elde edilirken, PSPNet ağı ile en düşük sonuçlar elde edilmiştir.

Çizelge 4.28: MVTec-Doku veri kümesinin test sonuçları.

Model	Ortalama Puanlar		Fold1		Fold2	
	mIoU	F1-Skoru	mIoU	F1-Skoru	mIoU	F1-Skoru
Swin-ÇÖEA	77,07	87,05	77,46	87,29	76,68	86,80
Swin-UNet	51,62	68,09	50,81	67,38	52,43	68,80
Attention-UNet	76,16	86,46	76,85	86,91	75,46	86,01
FPNet	74,51	85,39	75,36	85,95	73,65	84,83
LinkNet	65,21	78,91	62,61	77,00	67,80	80,81
PSPNet	64,93	78,73	66,18	79,65	63,67	77,80
U-Net	71,11	83,10	73,26	84,57	68,96	81,63

4.5.3 Tartışma ve analiz

Bu bölümde önerilen Swin-ÇÖEA mimarisinin bileşenlerini analiz etmek için bazı deneysel çalışmalar yapılmıştır. Ayrıca önerilen model literatürdeki en son yöntemler ile karşılaştırılarak güçlü ve zayıf yönleri ortaya konmuştur.

4.5.3.1 Önerilen ağdaki bileşenlerin analizi

Swin-ÇÖEA ağında dönüştürücü yapıları mekânsal detayları içeren öznetelikleri elde etmekte zayıf olduğu için [116, 118], kod çözücünün son aşamasında evrişim katmanı kullanılmıştır. Bu durumun geçerliliğini ve kod çözücünün son seviyesinde evrişim katmanı kullanımının etkisini göstermek için bazı deneysel çalışmalar yapılmıştır. Bu deneysel çalışmanın sonuçları Çizelge 4.29’da verilmiştir. Burada verilen Model-1’de kod çözücünün son seviyesinde Swin dönüştürücü kullanılmış ve Model-2’de ise EBR katmanı kullanılmıştır. Bu iki modelin elde ettiği sonuçlara göre, önerilen Model-2 MT ve MVTec veri kümeleri için sırası ile %81,37 ve %77,07 mIoU puanlarına ulaşırken, Model-1 sırası ile %79,90 ve %75,77 mIoU puanlarına ulaşmıştır. Bu sonuçlar karşılaştırıldığında kod çözücünün son seviyesinde EBR katmanı kullanılmasının etkili olduğu ispatlanmıştır.

Çizelge 4.29: Swin-ÇÖEA modelindeki kod çözücü katman analizi.

Önerilen model	Kod çözücünün son seviyesinde	mIoU (MT)	mIoU (MVTec)
Model-1	Swin dönüştürücü kullanma	79,90	75,77
Model-2	EBR kullanma	81,37	77,07

Yapılan ikinci analiz çalışmasında, önerilen modelin omurga ağ seçimi için farklı omurga ağları kullanılarak Swin-ÇÖEA modeli test edilmiştir. Bunun için önerilen modelde

kullanılan InceptionV3 ağının yanı sıra InceptionResNetv2, ResNext101, DenseNet169, EfficientNetb0, ResNext50, MobileNet, Vgg16 ve ResNet50 modelleri kullanılarak performans sonuçları elde edilmiştir. Bu performans sonuçları Çizelge 4.30’da verilmiştir.

Çizelge 4.30: Swin-MFNet modeli için farklı omurga ağları kullanılarak elde edilen sonuçlar.

MT			MVTec		
Model	mIoU	F1-Sıkor	Model	mIoU	F1-Sıkor
InceptionV3	81,37	89,73	InceptionV3	77,07	87,05
InceptionResNetv2	79,62	88,64	DenseNet169	76,83	86,90
DenseNet169	79,08	88,31	Vgg16	76,77	86,85
EfficientNetb0	79,06	88,28	ResNext50	76,41	86,62
ResNext50	78,76	88,11	EfficientNetb0	75,37	85,96
MobileNet	78,52	87,97	MobileNet	75,02	85,73
Vgg16	78,29	87,80	ResNet50	74,68	85,50
ResNet50	78,01	87,60	InceptionResNetv2	73,96	85,03

Çizelge 4.30’da görülebileceği gibi, MT ve MVTec veri kümeleri için en iyi skorlar (MT: %81,37 mIoU ve MVTec: %77,07 mIoU) InceptionV3 omurga ağı ile elde edilmiştir. Ek olarak, DenseNet169 modeli ortalama olarak ikinci en yüksek puana ulaşmıştır. Öte yandan, MT veri kümesi için ResNet50 mimarisi (%78,01 mIoU) ve MVTec veri kümesi için InceptionResNetv2 mimarisi (%73,96 mIoU) kullanılarak en düşük sonuçlar elde edilmiştir. Önceden eğitilmiş derin modellere dayalı deneysel çalışmalardan elde edilen sonuçlar (Çizelge 4.30) ele alınarak, önerilen Swin-MFNet modelinin omurga ağı için en iyi performansa sahip InceptionV3 mimarisi tercih edilmiştir.

4.5.3.2 Önerilen modelin önceki çalışmalarla karşılaştırılması

Bu çalışmada önerilen Swin-ÇÖEA modeli, MT ve MVTec-Doku veri kümelerine dayalı olarak daha önce yapılan çalışmaların performans puanları ile karşılaştırılmıştır. Bu karşılaştırmalı sonuçlar Çizelge 4.31’de verilmiştir. Bu sonuçlara göre Swin-ÇÖEA modeli MT ve MVTec-Doku veri kümelerinde en yüksek mIoU skoruna ulaşmıştır. İkinci en yüksek skorları MT’de DFF ile sağlanırken MVTec-Doku’da ortalama olarak CAVGA mimarisi ile sağlamıştır. Ayrıca MT’de FPNet ve dikkat kapıları ile geliştirilen PGA-Net mimarisi %71,31 mIoU ile en düşük puanı vermiştir. Öte yandan, otomatik kodlayıcıya dayalı olarak geliştirilen DPAE modeli, MT’de Swin-ÇÖEA modeline göre %11,73 daha düşük bir F1 skoru üretmiştir.

MVTec-Doku veri kümesi ile yapılan önceki çalışmalarda, genellikle otomatik kodlayıcı tabanlı anomali tespit yöntemleri kullanılmıştır. Çizelge 4.31’de verilen sonuçlara göre TVEVA, KRPCA-NTV, UTAD ve CAVGA yöntemlerinin mIoU sonuçları sırası ile %14,60, %54,29, %69,20 ve %73,40’tır. Otomatik kodlayıcılar görüntü yeniden oluşturmada oldukça etkili olmasına rağmen küçük yüzey hatalarını yakalamada zayıf kalmıştır. Sonuç olarak önerilen Swin-ÇÖEA mimarisi literatürde bulunan önceki çalışmalara göre MT ve MVTEC-Doku veri kümelerinin de daha üstün performanslar sağlamıştır.

Çizelge 4.31: MT ve MVTEC-Doku veri kümelerini kullanan önceki çalışmaların sonuçları.

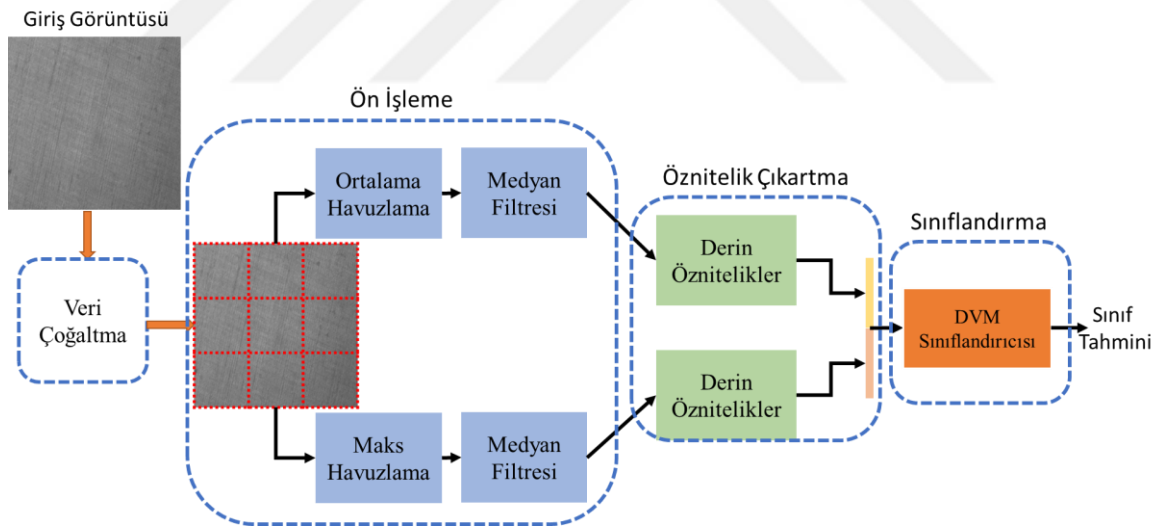
Referanslar	Yöntemler	Veri Kümesi	mIoU	F1-skoru
PGA-Net[7]	Vgg16, FPN, Dikkat Kapısı	MT	71,31	-
DPAE[140]	Dışbükey Olmayan Toplam Varyasyon, Otomatik Kodlayıcı		-	78,0
DFF[41]	Resnet50, Dikkat Kapısı		73,7	82,2
Swin-ÇÖEA	InceptionV3, Swin Dönüştürücü, KSMU		81,37	89,73
TVEVA[130]	Varyasyonlu Otomatik Kodlayıcı	MVTEC-Doku	14,60	-
KRPCA-NTV[74]	Dışbükey Olmayan Toplam Varyasyon, Otomatik Kodlayıcı		54,29	-
UTAD[157]	Gösterim Çıkarıcı, Otomatik Kodlayıcı		69,20	-
CAVGA[158]	Otomatik Kodlayıcı, Dikkat Kapısı		73,40	-
Swin-ÇÖEA	InceptionV3, Swin Dönüştürücü, KSMU		77,07	87,05

4.6 Uygulama 6: Derin Sinir Ağına Dayalı Çoklu Havuzlama ve Filtreleme Yaklaşımı

Yüzey hata tespitinde karşılaşılan zor problemlerden bir tanesi ürün yüzeyinde ya da dokusunda oluşan hataların küçük boyutlu ve belirsiz olmasıdır. Bu problem ile başa çıkmak için bu uygulamada Derin Sinir Ağına dayalı Çoklu Havuzlama ve Filtreleme (DSA-CHF) yaklaşımı önerilmiştir. Bu yeni yaklaşımda, öncelikle bir takım ön işleme ile yüzey hatası belirginleştirilmiştir. Daha sonra ResNet101 mimarisi kullanılarak derin öznitelikler çıkartılmıştır. Son olarak DVM sınıflandırıcısı ile doku hataları sınıflandırılmıştır. Bu uygulamada önerilen modelinin performans analizi için TILDA veri kümesi kullanılarak kapsamlı deneysel çalışmalar yapılmıştır.

4.6.1 Önerilen model

Önerilen DSA-CHF modelinin genel akış şeması Şekil 4.34'te ve uygulama adımları algoritma 4.3'te verilmiştir. Bu model, üç ana aşamadan oluşmaktadır: 1) ön işleme, 2) öznitelik çıkarma ve 3) sınıflandırma.



Şekil 4.34: Önerilen DSA-CHF modelinin genel şeması.

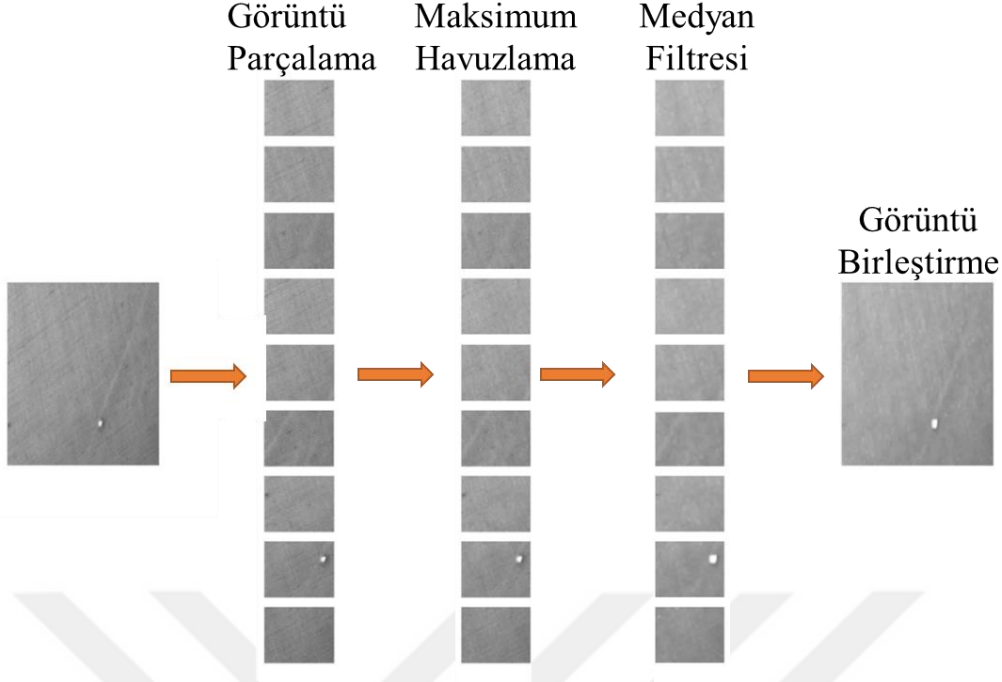
Algoritma 4.3: DSA-CHF modelinin uygulama adımları.

Girdi: Kumaş Görüntüsü
Çıktı: Kumaş hata tipi
1: Giriş görüntüsü $3 \times 3 = 9$ parçaya bölünmesi.
Durum 1 ve Durum 2 işlemlerinin uygulanması
<ul style="list-style-type: none">• Durum 1: Bölünmüş görüntünün her bir parçası için ortalama havuzlama ve ardından medyan filtresi uygulanması ve parçalarının birleştirilmesi.• Durum 2: Bölünmüş görüntünün her bir parçası için maksimum havuzlama ve ardından medyan filtresi uygulanması ve parçalarının birleştirilmesi.
2:
3: Bilinear enterpolasyon kullanarak doku görüntüsünü önceden eğitilmiş derin mimari yapısına göre yeniden boyutlandırılması.
4: Önceden eğitilmiş mimariyi kullanarak Durum-1 ve Durum-2'den elde edilen doku görüntüsü için derin özniteliklerinin çıkartılması.
5: Durum-1 ve Durum-2'den elde edilen derin özniteliklerinin birleştirilmesi.
6: Elde edilen öznitelikleri sınıflandırmak için DVM yöntemi uygulanarak sınıf etiketinin belirlenmesi.

Şekil 4.34 ve Algoritma 4.3'te, önerilen DSA-CHF modelinin detayları aşağıda alt başlıklar halinde verilmiştir.

4.6.1.1 Ön işleme aşaması

Ön işleme aşamasının amacı, dokuda var olan hataları netleştirmek ve böylece genel sınıflandırma performansını arttırmaktır. Bu amaçla öncelikle doku görüntüleri 9 parçaya ayrılmıştır. Daha sonra her parçaya (durum 1 için ortalama ve durum 2 için maksimum) havuzlama ve medyan filtresi uygulanmıştır. Son olarak parçalar birleştirilerek yeni görüntü oluşturulmuştur. Şekil 4.35'te, maksimum havuzlama ve medyan filtresine dayalı açıklayıcı bir örnek verilmiştir.



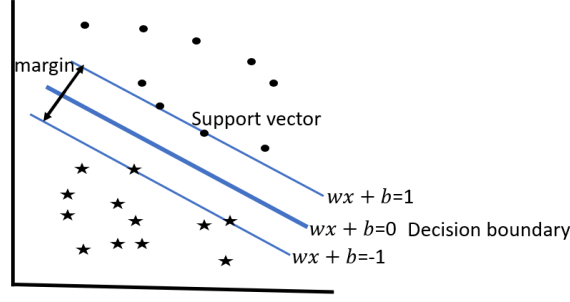
Şekil 4.35: Ön işleme aşamasında durum 1 için örnek gösterim.

4.6.1.2 Derin öznitelik çıkarma aşaması

DSA-ÇHF modelinde, önceden eğitilmiş ESA mimarisine dayalı iki farklı yaklaşım kullanılmıştır: 1) Derin öznitelik çıkarma, 2) İnce ayar (Fine-tuning) tabanlı transfer öğrenme. İlk yaklaşımda, önceden eğitilmiş derin mimarilerin tam bağlı katmanından derin öznitelikler çıkarılmıştır. Daha sonra bu öznitelikler kullanılarak bir DVM sınıflandırıcısı eğitilmiştir. İkinci yaklaşımda, önceden eğitilmiş derin mimarilerin son 3 katmanı yerine tam bağlı katman ve bir softmax katmanı eklenmiştir. Daha sonra bu mimari TILDA veri kümesi ile (ince ayar) eğitilmiştir.

4.6.1.3 Sınıflandırma aşaması

DSA-ÇHF modelinin sınıflandırma aşamasında, Cortes ve Vapnik [159] tarafından geliştirilen Destek Vektör Makinesi (DVM) yöntemi kullanılmıştır. DVM sınıflandırıcısının temel çalışma prensibi, iki sınıfı ayırabilecek en uygun doğruyu ($wx + b$) bulmaktır. Şekil 4.36’te görüleceği üzere $wx + b$ doğrusunu, iki sınıfı ayırabilecek şekilde tam ortada yerleştirmek için destek vektörleri kullanılır. Destek vektörleri iki sınıf arasındaki yakın bölgelerde yer alan verilerdir [92, 160, 161]. Daha detaylı bilgi için [92, 159, 160] nolu çalışmalara bakılabilir.



Şekil 4.36: DVM ile verilerin sınıflandırılması.

Bu çalışmada, SVM sınıflandırıcısı; uygulaması kolay olması, genelleme performansının yüksek olması ve yüksek boyutlu öznelik vektörlerinde etkili olması gibi nedenlerden dolayı tercih edilmiştir [92, 160, 161]. Bu yönteminin kutu kısıtlama düzeyi (box constraint level), çekirdek ölçeği (Kernel scale), ve çekirdek fonksiyonu (kernel function) gibi içerdiği sınıflandırıcı parametreleri performansı önemli ölçüde etkilemektedir. Mevcut çalışmada, deneme/yanılma yaklaşımı kullanılarak en iyi sınıflandırma performansına sahip parametreler seçilmiştir.

4.6.2 Deneysel çalışmalar

Önerilen Derin Sinir Ağına dayalı Çoklu Havuzlama ve Filtreleme (DSA-ÇHF) yaklaşımı, Intel Xeon Silver 2,19 GHz işlemci, NVIDIA P4000 Quadro GPU kartı ve 32 GB RAM ile bir bilgisayar kullanılarak gerçekleştirilmiştir. Ayrıca, önerilen ağ modellerini test etmek için 10 kat çapraz doğrulama yöntemi kullanılmıştır.

Deneysel çalışmada TILDA veri kümesinin e0-e6 sınıfları olan toplam 2800 görüntü kullanılmıştır (Çizelge 4.32). Ayrıca, çevirme ve yansıtma yöntemleri kullanılarak veri kümesindeki örnek sayısı arttırılmıştır. Orijinal veri kümesi ve veri arttırmaya dayalı olarak elde edilen örnek sayısı Çizelge 4.32’de verilmiştir.

Çizelge 4.32: Orijinal ve veri arttırma ile elde edilen veri kümeleri ve detayları.

Veri kümesi		Orijinal görüntü sayısı	Veri arttırma sonarı görüntü sayısı
C1	R1	350	2,100
	R3	350	2,100
C2	R2	350	2,100
	R3	350	2,100
C3	R1	350	2,100
	R3	350	2,100
C4	R1	350	2,100
	R3	350	2,100
Toplam		2,800	16,800

Deneysel çalışmalarda DVM sınıflandırıcısında Kuadratik/Kübik (Quadratic/Cubic) çekirdek işlevi ve bire karşı hepsi yaklaşımı (one-vs-all) kullanılmıştır. Derin öğrenme modellerinin eğitimi için batch boyutu 8 ve öğrenme oranı 0,0001 olarak ayarlanmıştır. Bunun yanı sıra transfer öğrenme için optimizasyon yöntemi olarak Olasılıksal Dereceli Azalma (Stochastic Gradient Descent) modeli kullanılmıştır.

4.6.3 Performans sonuçları

Önerilen DSA-CHF modelinin detaylı analizi için birçok farklı deneysel çalışma yapılmış ve performans sonuçları kaydedilmiştir. Bu performans sonuçları aşağıda başlıklar halinde detaylandırılmıştır.

4.6.3.1 Transfer öğrenme ve DVM sınıflandırıcı yaklaşımlarının analizi

Yapılan bu deneysel çalışmada sınıflandırıcı seçimi için iki yaklaşım analiz edilmiştir. Bu yaklaşımlar öznetelikleri DVM ile sınıflandıran yaklaşım ve ESA mimarisi kullanılarak transfer öğrenme tabanlı yaklaşımdır. Bu analiz için ResNet101 modelinden alınan öznetelikler kullanılmıştır. ResNet101 modelinden alınan öznetelikler transfer öğrenme (ince ayar) ve DVM sınıflandırıcı kullanılarak sınıflandırılmış ve elde edilen doğruluk puanları Çizelge 4.33'te verilmiştir. Çizelge 4.33'te verilen transfer öğrenme modelinde ResNet101 modelinden alınan öznetelikler tam bağlı katman ve sonrasında sınıflandırıcı için softmax katmanı kullanılmıştır.

Çizelge 4.33: DVM ve transfer öğrenme (ESA) tabanlı ResNet101 mimarisinin doğruluk puanları.

Veri kümesi		DVM	ESA (Transfer öğrenme)
C1	R1+ R3	81,14	75,71
C2	R2+ R3	79,14	72,86
C3	R1+ R3	80,71	62,86
C4	R1+ R3	67,86	60,71

Çizelge 4.33'ten de görüleceği üzere C1, C2, C3 ve C4 doku sınıfı için en yüksek doğruluk puanları DVM sınıflandırıcısı kullanılarak sırasıyla %81,14, %79,14, %80,71 ve %67,86 olarak elde edilmiştir. Daha da önemlisi, DVM'deki tüm sonuçların ESA'daki sonuçlardan daha yüksek olduğunu ve son katmanda bir DVM kullanmanın avantajlı olduğu görülmüştür.

4.6.3.2 TILDA veri kümesinden alınan detaylı sınıflandırma sonuçları

Bu bölümde, TILDA veri kümesi üzerinden detaylı deneysel çalışmalar yapılmıştır. Bu deneysel çalışmalarda TILDA veri kümesi 3 farklı yaklaşım ile ele alınmış ve bu yaklaşımlar ile önerilen DSA-CHF modelinin performans sonuçları kaydedilmiştir. Bunlar sırası ile şunlardır.

- Birinci yaklaşımda TILDA veri kümesinde bulunan tüm sınıfları ve alt sınıfları ayrı ayrı ele alınarak her sınıf ve alt sınıf için elde edilen sonuçlar,
- İkinci yaklaşımda her sınıfın alt sınıflar birleştirilerek C1, C2, C3 ve C4 sınıfları için elde edilen performans sonuçları,
- Üçüncü yaklaşımda TILDA veri kümesi bir bütün olarak değerlendirilmiş (genişletilmiş veri kümesi) ve bu genişletilmiş veri kümesi ile elde edilen sonuçlardır.

Bu deneysel çalışmalarda Bölüm 4.1’de detaylandırıldığı gibi DSA-CHF modeline DVM sınıflandırıcı ve ön işleme adımları dahil edilmiştir. Bu modelde DVM girişi için durum 1 ve durum 2 (bkz. Bölüm 4.1.1) ön işlemleri ile elde edilen netleştirilmiş görüntüler ve bu görüntülerden çıkartılan 2000 (her durum için 1000 öznitelik) adet derin öznitelik kullanılmıştır. Bunun yanı sıra derin öznitelikler çıkartmak için önceden eğitilmiş Resnet101 ağ mimarisi kullanılmıştır.

TILDA veri kümesinde, her sınıf (C1, C2, C3 ve C4) ve alt sınıfları (R1, R2 ve R3) için DSA-CHF modeli kullanılarak elde edilen performans sonuçları Çizelge 4.34’te verilmiştir.

Çizelge 4.34: Önerilen DSA-CHF modelinin bireysel doğruluk puanları.

Veri kümesi	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	
C1	R1	98,52	98,54	98,52	98,52
	R3	98,14	98,17	98,14	98,14
C2	R2	97,24	97,26	97,24	97,24
	R3	98,67	98,68	98,67	98,67
C3	R1	96,43	96,45	96,43	96,43
	R3	94,71	94,81	94,71	94,72
C4	R1	94,33	94,40	94,33	94,31
	R3	95,81	95,82	95,81	95,81

Çizelge 4.34'te verilen sonuçlara göre en yüksek doğruluk puanı %98,67 (C2-R3 sınıfında) olarak elde edilirken, en düşük doğruluk puanı %94,33 (C4-R1 sınıfında) olarak elde edilmiştir. Genel olarak, C1-R1, C1-R3, C2-R2, C2-R3, C3-R1 ve C4-R3 sınıfları için önerilen model %95'in üstünde bir doğruluk skoru sağlamıştır.

TILDA veri kümesinden alınan ikinci bireysel performans sonuçları, birleştirilmiş veri kümesi kullanılarak elde edilmiştir. Bu veri kümesinde her C sınıfı için alt R sınıfları birleştirilerek bir bütün olarak ele alınmıştır. Bu veri kümesinde, önerilen model ile elde edilen performans sonuçları Çizelge 4.35'te verilmiştir.

Çizelge 4.35: Önerilen DSA-CHF modelinin birleştirilmiş veri kümeleri üzerindeki doğruluk puanları.

Veri kümesi	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	
C1	R1+ R3	97,36	97,39	97,36	97,36
C2	R2+ R3	97,74	97,76	97,74	97,74
C3	R1+ R3	95,93	95,96	95,93	95,93
C4	R1+ R3	94,83	94,83	94,83	94,83
C1+ C2+ C3+ C4		95,82	95,84	95,82	95,82

Çizelge 4.35'e göre, en iyi doğruluk puanı %97,74 ile C2 sınıfında elde edilmiştir. İkinci en iyi skoru ise %97,36 ile C1 sınıfında alınmıştır. Öte yandan C3 ve C4 sınıfları için doğruluk puanları sırasıyla %95,93 ve %94,83 olarak kaydedilmiştir.

TILDA veri kümesinden alınan üçüncü bireysel performans sonuçları genişletilmiş veri kümesi kullanılarak elde edilmiştir. Bu veri kümesinde TILDA'nın dört sınıfı (C1, C2, C3

ve C4) birleştirilmiştir. Bu veri kümesi ile yapılan deneysel çalışmada önerilen model, %95,82'lik doğruluk puanına ulaşmıştır. Genişletilmiş TILDA veri kümesi için önerilen model kullanılarak elde edilen en iyi performans sonuçlarının karışıklık matrisleri Şekil 4.37'da gösterilmiştir.

Gerçek Değer	e0	2320	5	19	12	13	6	25	96.7%	3.3%
	e1	15	2255	25	79	17	4	5	94.0%	6.0%
	e2	37	17	2282	16	37	1	10	95.1%	4.9%
	e3	23	46	15	2298	9	1	8	95.8%	4.2%
	e4	41	27	42	13	2265	6	6	94.4%	5.6%
	e5	10	5	6	2	3	2365	9	98.5%	1.5%
	e6	46	10	10	10	8	4	2312	96.3%	3.7%
		e0	e1	e2	e3	e4	e5	e6	Tahmin	

Şekil 4.37: Önerilen modelin karışıklık matrisleri.

Şekil 4.4, önerilen DSA-ÇHF modelini kullanarak genişletilmiş veri kümesi için e0 (normal), e1, e2, e3, e4, e5 ve e6 doku hatalarının sınıflandırma performanslarını göstermektedir. Bu sonuçlara göre önerilen model ile e0 olarak gösterilen hatasız (normal) doku görüntüleri için %96,7 doğruluk puanı elde edilmiştir. Ayrıca e1, e2, e3, e4, e5 ve e6 doku hataları için elde edilen en yüksek doğruluk puanı e5 doku hatası (%98,5) olmuştur.

4.6.3.3 Son teknoloji modeller ile karşılaştırma

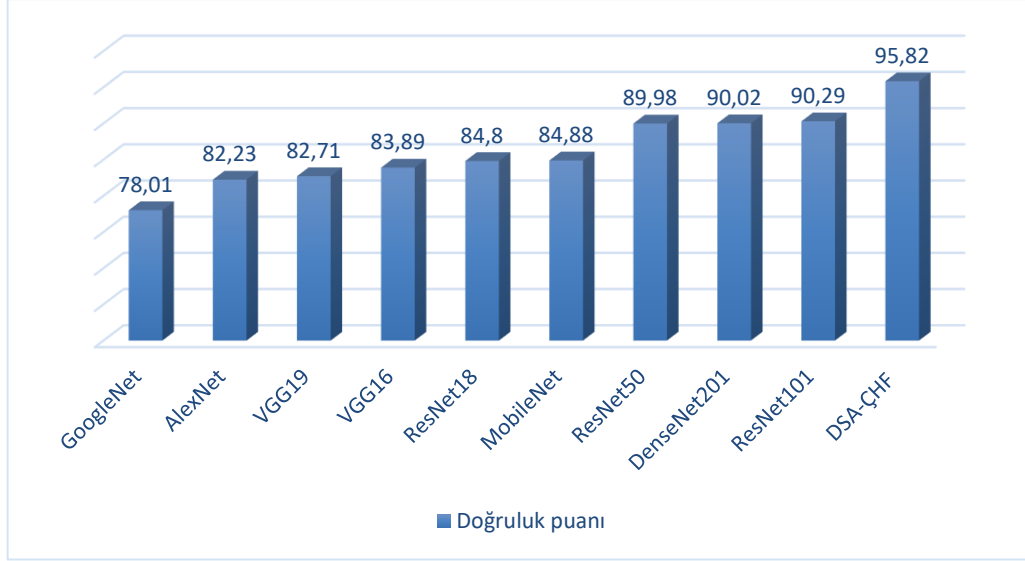
Son zamanlarda sınıflandırma için birçok farklı ESA mimarisi geliştirilmiştir. Bu bölümde, önerilen DSA-ÇHF modeli ile son teknolojik ESA mimarilerinin performansları karşılaştırılmıştır. Bu deneysel çalışmalarda önceden eğitilmiş ESA mimarilerinde elde edilen öznitelikler DVM sınıflandırıcı kullanılarak eğitilmiştir. Elde edilen sayısal sonuçlar Çizelge 4.36'da verilmiştir.

Çizelge 4.36: Önerilen DSA-ÇHF modelinin diğer mimarilerle karşılaştırılması.

Model	Doğruluk puanları							
	Veri arttırma olmadan				Veri arttırma ile			
	C1	C2	C3	C4	C1	C2	C3	C4
AlexNet	75,71	69,43	71,86	55,14	88,64	85,76	86,33	76,40
MobilNet	76,86	73,86	68,86	61,86	93,36	91,86	87,19	86,88
GoogleNet	70,57	68,14	68,71	53,86	87,24	82,19	81,88	78,64
VGG16	76,14	77,57	76,86	65,86	88,05	88,14	86,71	80,55
VGG19	79,57	73,71	74,71	62,14	89,83	87,69	86,33	79,33
ResNet18	75,29	69,57	68,57	61,86	90,24	90,64	83,52	85,83
ResNet50	80,57	77,43	77,71	68,29	94,21	94,76	92,71	90,60
ResNet101	81,14	79,14	80,71	67,86	94,55	95,24	93,36	91,45
DenseNet201	83,14	79,29	78,43	65,14	95,33	95,71	92,83	89,74
DSA-ÇHF	83,73	84,86	83,86	71,57	97,36	97,74	95,93	94,83

Çizelge 4.36’da önerilen DSA-ÇHF modelinin performansı, dört (C1, C2, C3 ve C4) sınıf için mevcut önceden eğitilmiş derin mimarilerle karşılaştırılmıştır. Bunun yanı sıra, veri arttırma yöntemleri kullanılarak performans sonuçları verilmiştir. Bu sonuçlara göre, veri arttırmaya dayalı olarak elde edilen veri kümesini kullanan tüm derin modeller, orijinal veri kümesini kullanılan (veri arttırmaz) derin modellere kıyasla yaklaşık %13-18’lik bir performans artışı elde edilmiştir. Ayrıca, verisi arttırılmış veri kümesi için elde edilen sonuçlar dikkate alındığında, önerilen DSA-ÇHF modelinin, dört sınıfın tamamında mevcut diğer önceden eğitilmiş modellere kıyasla daha üstün bir doğruluk performansı ürettiği gözlemlenmiştir. C1 ve C2 sınıfları için mevcut diğer derin mimariler açısından, en yüksek doğruluk puanı DenseNet201 mimarisi kullanılarak elde edilirken, C3 ve C4 sınıfları için en yüksek doğruluk puanı ResNet101 mimarisi kullanılarak elde edilmiştir. Derin öğrenme mimarileri arasında ortalama olarak en düşük performans ise GoogleNet mimarisi ile elde edilmiştir.

Önerilen DSA-ÇHF modelinin ve diğer mevcut önceden eğitilmiş modellerin, genişletilmiş veri kümesi (tüm sınıflar dahil) kullanılarak karşılaştırmalı performansları Şekil 4.38’de sunulmuştur.



Şekil 4.38: DSA-CHF modeli ve diğer ESA mimariler ile karşılaştırması.

Şekil 4.38’de görülebileceği üzere, önerilen model, en yüksek doğruluk puanına sahip olan ResNet101 mimarisine kıyasla dört sınıfın tümü (C1, C2, C3 ve C4) için yaklaşık %5’lik önemli bir performans artışı sağlamıştır. Derin mimariler arasında ikinci en iyi puan DenseNet201 mimarisi ile %90,02 iken, en kötü performans GoogleNet kullanılarak %78,01 olarak elde edilmiştir. ResNet50, VGG16, MobileNet, VGG19, ResNet18 ve AlexNet mimarileri ile doğruluk puanları sırasıyla %89,98, %83,89, %84,88, %82,71, %84,8 ve %82,23 olarak elde edilmiştir.

4.6.4 Tartışma ve analiz

Bu bölümde önerilen DSA-CHF mimarisinin bileşenlerini analiz etmek için bazı deneysel çalışmalar yapılmıştır. Ayrıca önerilen model literatürdeki en son yöntemler ile karşılaştırılarak güçlü ve zayıf yönleri ortaya konmuştur.

4.6.4.1 Önerilen modelin ön işleme adımının analizi

DSA-CHF modelinde, doku hatalarını tespit etmek için sınıflandırma ve öznetelik çıkarma aşamalarından önce ön işlem aşamasında bir takım yeni yaklaşımlar uygulanmıştır. Bu yaklaşımın analizi için aşağıdaki deneysel çalışmalar yapılmıştır. Tüm deneysel çalışmalarda genişletilmiş veri kümesi kullanılmıştır.

Birinci deneysel çalışmada, ön işleme aşamasında bölme işlemi kullanılmadan ortalama havuzlama + medyan filtresi (median filter) ve maksimum (max) havuzlama + medyan filtresi işlemleri ayrı ayrı uygulanmıştır. Ön işlem aşamasından sonra elde edilen görüntüler için derin ESA modeli kullanılarak derin öznetelikler çıkarılmış ve ardından DVM yöntemi

kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Bu deneysel çalışmalardan elde edilen bireysel ve birleştirme sonuçları Çizelge 4.37’de verilmiştir.

Çizelge 4.37: Birinci deneysel çalışmanın doğruluk puanları.

Ön İşleme Aşaması	Derin Öznitelik sayısı	Doğruluk Puanı
Ortalama havuzlama + medyan filtre	1000	%89,12
Maksimum havuzlama + medyan filtre	1000	%89,51
Birleştirme (ortalama havuzlama + medyan filtre / maksimum havuzlama + medyan filtre)	2000	%93,37

Çizelge 4.37’de verilen sonuçlara göre doku hatasını sınıflandırmak için ortalama havuzlama + medyan filtre işlemleri kullanılarak elde edilen doğruluk puanı %89,12 iken, maksimum havuzlama + medyan filtre işlemi ile elde edilen doğruluk puanı %89,51’dir. Öte yandan bu iki işlem birleştirilerek doğruluk puanı %93,37 olarak elde edilmiştir. Ayrıca ön işleme yapılmadan ResNet101 mimarisi ve DVM sınıflandırıcısı ile %90,29 doğruluk (Şekil 4.5) puanı elde edilmiştir. Ortalama havuzlama + medyan filtre veya maksimum havuzlama + medyan filtre işlemi kullanıldığında bu performans yaklaşık %1 oranında azaldığı görülmüştür. Bu düşük performansın nedeni, bazı görüntülerde doğrudan ön işlemler kullanıldığında, görüntülerdeki küçük hataların daha görünmez hale gelmesidir. Bu problemi çözmek için ön işleme aşamasının başında doku görüntüleri $n \times n$ parçaya bölünmüştür. Daha sonra, bölünmüş görüntülerin her biri için ortalama havuzlama + medyan filtre ve maksimum havuzlama + medyan filtre işlemleri uygulanmıştır. Bu bölme işlemi sayesinde görüntülerdeki küçük hatalar daha belirgin ve net hale gelmiştir. Son olarak, derin ESA modeli kullanılarak derin öznitelikler çıkarılmış ve daha sonra DVM yöntemi ile sınıflandırma işlemi gerçekleştirilmiştir. Bu deneysel çalışmalardan elde edilen bireysel ve birleştirme sonuçları Çizelge 4.38’de gösterilmiştir. Bu deneysel çalışmada 2x2, 3x3, 4x4 ve 5x5 bölme yöntemleri kullanılmış ve en iyi performans 3x3 (9 adet) bölme yaklaşımı kullanılarak elde edilmiştir.

Çizelge 4.38: Üçüncü deneysel çalışmanın doğruluk puanları.

Ön İşleme Aşaması	Derin Öznitelik sayısı	Doğruluk Puanı
Ortalama havuzlama + medyan filtre	1000	%93,12
Maksimum havuzlama + medyan filtre	1000	%93,22
Birleştirme (ortalama havuzlama + medyan filtre / maksimum havuzlama + medyan filtre)	2000	%95,82

Çizelge 4.38’de verilen sonuçlara göre bölme işlemine dayalı ortalama havuzlama + medyan filtre işlemleri kullanılarak elde edilen doğruluk puanı %93,12 iken, bölme işlemine dayalı maksimum havuzlama + medyan filtre işlemi ile elde edilen doğruluk puanı %93,22’dir. Bu işlemlerden elde edilen bireysel performanslar, orijinal Resnet101 modeline göre %3’lük bir performans artışı sağlamıştır. Öte yandan bu iki işlem birleştirilerek doğruluk puanı %95,82 olarak elde edilmiştir. Bu sonuçlara göre önerilen bölme işlemine dayalı ortalama/maksimum havuzlama + medyan filtre ve derin öznitelikler yaklaşımı, orijinal Resnet101 modeline göre %5 performans artışı sağlamıştır.

4.6.4.2 Önceki çalışmalar ile kıyaslanması

Literatürde doku sınıflandırmasına dayalı olarak birçok çalışma bulunmaktadır. Bu çalışmalarda genellikle yazarlar tarafından oluşturulmuş bireysel veri kümeleri kullanılmıştır. Bunun yanı sıra literatürde kumaş hata tespit için en yaygın olarak uygulanan TILDA veri kümesidir. TILDA veri kümesini kullanan mevcut çalışmaların doğruluk puanları Çizelge 4.39’da verilmiştir.

Çizelge 4.39: Önerilen model sonuçlarının önceki çalışmalarla karşılaştırılması.

Referanslar	Yöntemler	Veri kümesi	Sınıf ve görüntü sayısı	Doğruluk Puanı
[45]	İnce ayarlı AlexNet	C1-R1	6 sınıf ve 300 görüntü	%97,2
[37]	LBP, GLCM ve Sinir Ağı	C1-R1 ve C1-R2	6 sınıf ve 600 görüntü	%97,6
[162]	LBP, GLCM ve DVM	-	7 sınıf ve 480 görüntü	%86,7
[38]	GLCM, LBP, LPQ ve DVM	-	5 sınıf ve 500 görüntü	%97,25
[163]	ResNet512 tabanlı ESA öznelikleri, Kullback Leibler Divergence (KLD) ve Markov Random Field (MRF)	C1-R2, C2-R2	6. sınıf	%98,5 ve %96,5
[164]	AlexNet tabanlı Çoklu ölçekleme derin ESA	-	6 sınıf ve 1850 görüntü	%96,55.
[39]	GLCM, Gabor Dalgacık ve Rastgele Karar Ormanı	-	6. sınıf	%84,5
Önerilen DSA-CHF model	Derin Sinir Ağına dayalı Çoklu Havuzlama ve Filtreleme yaklaşımı	C1 (R1,R3), C2 (R2,R3), C3 (R1,R3) ve C4 (R1,R3)	7 sınıf ve 2800 görüntü	97,36, 97,74, 95,93 ve 94,83

Çizelge 4.39’da önerilen model, TILDA veri kümesini kullanan önceki çalışmaların doğruluk puanları ile karşılaştırılmıştır. Bu verilere göre, önerilen yaklaşımın sınıf sayısı ve görüntü sayısı dikkate alındığında mevcut çalışmalara göre daha başarılı olduğu görülmüştür. Ayrıca önerilen yaklaşım, TILDA veri kümesindeki tüm sınıfları ve görüntüleri kullanan en kapsamlı çalışma olarak görünmektedir. Tüm bunlara ek olarak, [37], [162], [38], ve [39] nolu çalışmalarda geleneksel öznelilik çıkarma yaklaşımları kullanılmıştır. Bu çalışmalar arasında en yüksek doğruluk %97,6 ile [37] nolu çalışmada elde edilmiştir. Öte yandan, [164], [163] ve [45] nolu çalışmalarda AlexNet ve ResNet512 derin modeller kullanılmıştır. Bu çalışmalar arasında en yüksek doğruluk %98,5 ile [163] nolu çalışmada elde edilmiştir. Tüm bu sonuçlara göre sınıf sayısı ve görüntü sayısı dikkate alındığında derin modellerin geleneksel yöntemlere göre daha başarılı olduğu belirlenmiştir.

Çizelge 4.39’da verilen bu çalışmalarda TILDA veri kümesinin belirli bir kısmı kullanılmıştır. Mevcut çalışmada, TILDA veri kümesindeki tüm sınıflar ve görüntüler

kullanılarak kapsamlı deneysel çalışmalar yapılmıştır. Bu nedenle, önerilen DSA-CHF modeli ile mevcut çalışmalar arasında performans sonuçlarının gerçek bir karşılaştırması yanlış ve yanıltıcı olacaktır. Ancak önerilen model, TILDA veri kümesindeki tüm sınıflar ve görüntüler kullanılarak test edilmiş ve daha az sınıf ve görüntüye dayalı mevcut çalışmalara göre daha üstün bir başarı elde ettiği söylenebilmektedir.



5. SONUÇLAR VE ÖNERİLER

Çelik, kumaş ve mermer gibi imalat sistemlerinde yüzey hatalarının tespit edilmesi, maddi kayıpları önlenmesi ve yüksek kaliteli ürün sağlanması açısından oldukça önemlidir. Yüzey hataları olmayan yüksek kaliteli ürünleri tedarik eden üreticiler tüketicinin güvenini kazanacaktır. Bu sayede günümüzde birçok imalat sistemlerinin içinde bulunduğu zorlu rekabet karşısında avantajlı olacaktır. Bu nedenle tekstil, mermer ve çelik gibi endüstriyel sektörlerdeki yüzey hatalarının otomatik olarak tespit edilmesi büyük önem kazanmıştır.

Bu tez çalışması kapsamında, otomatik yüzey hata tespiti için derin öğrenme tabanlı özgün yöntemler geliştirilmiştir. Bu yöntemlerde günümüzde oldukça popüler olan ESA mimarileri, sıkma-uyarlama dikkat kapıları ve dönüştürücüler gibi güncel modüller kullanılmıştır. Bu doğrultuda 6 farklı özgün derin öğrenme tabanlı model önerilmiştir. Bu yöntemler otomatik yüzey hata tespitinde yüksek performanslar elde ettiği gözlemlenmiştir.

Uygulama 1’de, piksel seviyesinde yüzey hata tespiti için InceptionV3 tabanlı Zenginleştirilmiş Öznitelik Entegrasyon ve Anlamlandırma Ağ (Inc-ZÖEA) mimarisi önerilmiştir. Bu mimari, önceden eğitilmiş modellerden daha detaylı öznitelikler elde edilmesini ve hata tespit performansının artırılmasını amaçlamıştır. Bu doğrultuda, önceden eğitilmiş InceptionV3 ağ mimarisinin her seviyedeki katmanlarından öznitelikler çıkartılmıştır. Daha sonra elde edilen özniteliklerden aynı yükseklik ve genişlikteki öznitelik haritaları birleştirilmiştir. Birleştirilen öznitelik haritaları arasında önemli detayları güçlendirmek için MSKU bloğu kullanılmıştır. Bu işlemler sayesinde önceden eğitilmiş InceptionV3 mimarisinden detaylı öznitelikler çıkartılmış ve önemli ayrıntılar güçlendirilmiştir. Önerilen modelin ikinci aşamasında, elde edilen güçlendirilmiş özniteliklerden piksel seviyesinde hata tespit tahmin haritası üretilmiştir. Bu amaç doğrultusunda, önerilen modelin kod çözücü birimi olarak görev alan Öznitelik Entegrasyon ve Anlamlandırma modülünde öznitelikler bir noktada birleştirilmiştir. Bu öznitelik haritasında hata tespitinde en önemli mekânsal ve anlamsal bilgileri ortaya çıkarmak/güçlendirmek için MKSU bloğu uygulanmıştır. Son olarak Inc-ZÖEA mimarisinin çıktısında sigmoid fonksiyonu ile piksel seviyesinde hata tespiti yapılmıştır. Önerilen Inc-ZÖEA mimarisi DAGM, MVTec-Doku ve MT veri kümeleri ile yapılan deneysel çalışmalarda üstün performans sergilediği kanıtlanmıştır. Ayrıca Inc-ZÖEA yöntemi son teknoloji yöntemler ile karşılaştırılmış ve analizler yapılmıştır. Sonuç olarak,

MT, MVTEC-Doku ve DAGM veri kümelerinde sırası ile Inc-ZÖEA mimarisi %77,44 mIoU, %81,2 mIoU ve %79,46 mIoU ile en yüksek performansları elde etmiştir.

Uygulama 2’de, piksel seviyesinde yüzey hata tespiti için MKSU blok yapısını temel olarak yeni bir dikkat blok yapısı önerilmiştir. Derinlemesine Sıkma ve Uyarlama Bloğu (DSUB) olarak adlandırılan bu yapı MKSU blok yapısında kullanılan noktasal evrişim katmanı yerine iki aşamadan oluşan derinlemesine ayrılabilir evrişim katmanı kullanılmıştır. Bu yapı sayesinde önemli öznitelikleri ortaya çıkartan ağırlıkları elde etmek için daha etkili bir yaklaşım sunulmuştur. Yapılan bileşen analizlerinde önerilen DSUB bloğu MKSU bloğundan daha etkili olduğu kanıtlanmıştır. Ayrıca yapılan çalışmada DSUB-EUNet (Derinlemesine Squeeze and Excitation Block tabanlı Efficient UNet) mimarisi önerilmiştir. Önerilen bu mimaride, UNet ağının kodlayıcı ve kod çözücü atlama bağlantıları arasına DSUB yapıları eklenmiştir. Deneysel çalışmalarda önerilen model ve bileşenlerinin analizleri için kapsamlı testler gerçekleştirilmiştir. Öte yandan önerilen DSUB-EUNet modeli, yüzey hata tespiti için üç farklı veri kümesi kullanılarak test edilmiştir. Deneysel sonuçlara göre, önerilen DSUB-EUNet ağı, LinkNet, PSPNet, FPNet, U-net, VGG-UNet, Mobil-UNet, Eff-UNet, Attention-UNet, Trans-UNet, ve Swin-UNet gibi son teknoloji yöntemlerden ve önceki çalışmalardan daha iyi performans üretmiştir. (MT: %83,19 mIoU ve %90,82 F1-Skoru, DAGM: %82,31 mIoU ve %90,29 F1-Skor; AITEX: %86,69 mIoU ve %92,87 F1-Skoru).

Uygulama 3’te yüzey hata tespitinde karşılaşılan sınırlı veri kümesi problemi ele alınmıştır. Bu problem temel olarak, hatalı görüntülerin elde edilmesi zaman alıcı ve maliyetli olduğu için genellikle sınırlı hatalı yüzey görüntüsü içeren veri kümelerinden oluşmasıdır. Bu problemde yola çıkarak daha az parametreye sahip ve etkili bir ağ mimarisi geliştirilmesi hedeflenmiştir. Bu hedef doğrultusunda derinlemesine ayrılabilir evrişim katmanlarından oluşan ve MFE-DEDNet olarak adlandırılan yeni bir ağ mimarisi tasarlanmıştır. Ayrıca bu mimaride global güçlü mekânsal ve spektral öznitelikler elde etmek için 3D Spektral ve Mekânsal öznitelik çıkarma (3BSM) modülü geliştirilmiştir. MFE-DEDNet olarak adlandırılan bu ağ mimarisi sadece 5,1 milyon parametre içermesine rağmen etkili sonuçlar vermiştir. Önerilen mimarinin kodlayıcı kısmı, derinlemesine ayrılabilir evrişim katmanlarından oluşan MÖÇ ve 3BSM modüllerinden oluşmaktadır. Bu modüller sayesinde güçlü derin öznitelik haritaları elde edilmiştir. MFE-DEDNet ağının kod çözücü bölümünde, bu öznitelik haritaları, çoklu giriş dikkat kapısı olarak görev alan ÇGDK modülü kullanılarak etkili bir şekilde birleştirilmiştir. Önerilen model, MT ve MVTEC veri kümeleri

kullanılarak detaylı performans ve bileşen analizleri yapılmıştır. Elde edilen sonuçların doğruluğunu test etmek için 4 kat çapraz doğrulama modeli kullanılmıştır. Önerilen MFE-DEDNet ağı kullanılarak, MT ve MVTec veri kümelerinde üstün başarılar elde edilmiştir (MT: %80,01 mIoU, %88,86 F1-sıkor ve %99,07 AUC; tüm MVTec sınıflarının ortalaması: %56,81 mIoU, %69,83 F1-sıkor ve %95,99 AUC). Bunun yanı sıra bileşen analizinde önerilen 3BSM modülünün etkili olduğu kanıtlanmıştır. Sonuç olarak, önerilen modelin etkili ve daha az sayıda parametre kullanılarak önceki çalışmalara kıyasla daha yüksek performansa sahip olduğu gözlenmiştir.

Uygulama 4'te son zamanlarda oldukça popüler olan dönüştürücü modellerini yüzey hata tespitine uygulamak için yeni bir dikkat yapısı geliştirilmiştir. Melez Dikkat Kapısı (MDK, Hybrid Attention Gate) olarak adlandırılan bu yapıda ViT ve KSMU yapılarından oluşmaktadır. Bölüm 4.5'te detaylandırılan bu uygulamada Önerilen MDK yapısını yüzey hata tespitine uygulamak için MDK tabanlı Öznitelik Füzyon Ağı (MDK-ÖFA) dayalı yeni bir ağ mimarisi tasarlanmıştır. MDK modülünde, önceden eğitilmiş MobileNetV2 mimarisinden elde edilen başlangıç özniteliklerine tasarlanan Trans (dönüştürücü) ve KSMU blok yapıları paralel olarak uygulanmıştır. Trans modelinde öz-dikkat mekanizması ile küresel anlamsal öznitelikler elde edilirken, KSMU modelinde mekânsal öznitelikler elde edilmiştir. Ardından, bu iki modelin çıktıları birleştirilmiştir. Önerilen MDK modülünde, Trans ve KSMU çıktılarını etkili bir şekilde birleştirmek için concat (doğrudan birleştirme), add (eklemeli birleştirme), max (maksimum ile seçerek birleştirme) ve mul (çarpma işlemi ile birleştirme) modelleri geliştirilmiştir. Deneysel çalışmalarda MT ve MVTec-Doku veri kümesi kullanılarak önerilen birleştirme modellerinin etkili sonuçlar verdiği kanıtlanmıştır. Elde edilen sonuçlarda MDK^{concat} -ÖFA, MDK^{add} -ÖFA, MDK^{max} -ÖFA ve MDK^{mul} -ÖFA modelleri arasında MDK^{max} -ÖFA ve MDK^{concat} -ÖFA modelleri ile hata tespiti için en yüksek skorlar sağladığı gözlemlenmiştir. Öte yandan MDK^{concat} -ÖFA modeli ile küçük yüzey hatalarının tespitinde daha başarılı sonuçlar sağladığı tespit edilmiştir. Ayrıca yapılan detaylı deneysel çalışma ve analizler sonucunda MDK^{max} -ÖFA modeli yüzey hata tespiti için etkili olduğu ve Trans-UNet, Swin-UNet, Attention-UNet, FPNet, LinkNet, PSPNet, UNet gibi son teknoloji modellere göre daha üstün performans sağladığı gözlemlenmiştir.

Uygulama 5'te yakın zamanlarda geliştirilen Swin-dönüştürücü (Swin-Transformer) modeli ele alınmıştır. Bu uygulamada Swin-dönüştürücü ile geliştirilen Swin-UNet ağına kıyasla yüzey hata tespiti için çok daha başarılı sonuçlar veren Swin-ÇÖEA adlı yeni bir Swin-dönüştürücü tabanlı ağ mimarisi önerilmiştir. Önerilen model, InceptionV3, Swin

Dönüştürücü, geleneksel evrişim ve KSMU yapılarının etkili bir şekilde birleşiminden oluşmaktadır. İlk olarak, yüzey hata tespiti için Swin-UNet'in aksine önerilen modelde InceptionV3 mimarisiyle güçlü başlangıç öznelikleri çıkartılmıştır. Bu sayede yetersiz veri kümelerinde zayıf kalan swin dönüştürücüleri bloğunu beslemek için güçlü öznelıklar elde edilmiştir. Öte yandan önerilen Swin-ÇÖEA ağında, Swin-Unet'ten farklı olarak önerilen kod çözücü bölümünde Swin dönüştürücü ve geleneksel evrişim tabanlı hibrit bir model geliştirilmiştir. Evrişim işleminin kullanılmasındaki temel amaç dönüştürücülerin mekânsal ayrıntıları yakalamada zayıf olduğu için evrişim katmanları ile desteklenmesidir. Yapılan bileşen analizlerinde, bu yaklaşımın yüksek performanslar sağladığı kanıtlanmıştır. Yapılan kapsamlı deneysel çalışmalarda önerilen Swin-ÇÖEA modeli MT ve MVTec-Doku veri kümeleri için mIoU puanları sırasıyla %81,37 ve %77,07 olarak elde edilmiştir. Elde edilen sonuçlara göre önerilen modelin dönüştürücü tabanlı Swin-Unet ve Trans-Unet modelinden daha üstün olduğu kanıtlanmıştır.

Uygulama 6'da doku hatalarının sınıflandırılması için Derin Sinir Ağı tabanlı Çoklu Havuzlama ve Filtreleme (DSA-ÇHF) yaklaşımı önerilmiştir. Önerilen modelin temel amacı doku görüntülerinde oluşan hataların belirginleştirilerek küçük ve belirsiz yüzey hatalarını ortaya çıkartılmasıdır. Ayrıca önerilen modelde güçlü derin öznelıklar elde edilerek sınıflandırma performansının artırılması hedeflenmiştir. Gerçekleştirilen DSA-ÇHF modelinde, uygulanan ortalama/maksimum havuzlama ve medyan filtresi tabanlı ön işlem sayesinde yüzey hataları daha belirgin olmuştur. Ön işleme aşamasında öncelikle doku görüntüleri 3x3 eşit parçaya bölünmüştür. Ardından her parçaya medyan filtreleme ve havuzlama işlemleri uygulanmıştır. Bölüm 4.1'de verilen ön işleme analizlerinde önerilen ön işlemin doku hatalarını belirginleştirerek performansı arttırdığı gözlemlenmiştir. Öte yandan önerilen modelde önceden eğitilmiş ResNet101 mimarisi ile görüntülerden derin öznelıklar çıkartılmış ve DVM sınıflandırıcısı ile doku hataları sınıflandırılmıştır. Yapılan kapsamlı analizlerde, Resnet101 ve DVM sınıflandırıcısı tabanlı önerilen model, diğer derin öğrenme modelleri karşısında üstün performans sergilediği gözlemlenmiştir. Önerilen DSA-ÇHF modelini test etmek için çok sınıflı TILDA veri kümesi kullanılmıştır. Kapsamlı deneysel çalışmalarda, önerilen DSA-ÇHF model, C1, C2, C3 ve C4 sınıfları için sırasıyla %97,36, %97,74, %95,93 ve %94,83 doğruluk skorları üretmiştir. Ayrıca, dört sınıflı birleştirerek yapılan deneysel çalışmada %95,82 doğruluk skoru elde edilmiştir.

Tez çalışması süresince yapılan çalışmalar ve elde edilen bulgulara dayalı olarak gelecekte yapılması düşünülen çalışmalar şunlardır: 1) Online öğrenebilen sistemler

geliştirilerek sürekli güncel kalacak otomatik yüzey hata tespit sistemleri geliştirilebilir. 2) Mobil tabanlı otomatik yüzey hata tespit yazılımları geliştirilerek donanımdan bağımsız ve taşınması kolay bir sistem sağlanabilir.



KAYNAKLAR

- [1] **Kaya, İ., Oktay, S., & Engin, O.** (2005). Kalite Kontrol Problemlerinin Çözümünde Yapay Sinir Ağlarının Kullanımı. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 21(1), 92–107. Retrieved from <https://dergipark.org.tr/en/pub/erciyesfen/issue/25601/270159>
- [2] **Kesici, B., & Yıldız, M. S.** (2016). Kalite Kontrol Faaliyetlerinde Yapay Zekâ Kullanımı ve Bir Otomotiv Yan Sanayisinde Uygulanması. *Yalova Üniversitesi Sosyal Bilimler Dergisi*, 6(12), 307–323. Retrieved from <https://dergipark.org.tr/en/pub/yalovasosbil/issue/27392/289024>
- [3] **Asoudegi, E., & Pan, Z.** (1991). Computer vision for quality control in automated manufacturing systems. *Computers & Industrial Engineering*, 21(1–4), 141–145. [https://doi.org/10.1016/0360-8352\(91\)90078-K](https://doi.org/10.1016/0360-8352(91)90078-K)
- [4] **Tao, X., Zhang, D., Ma, W., Liu, X., & de Xu.** (2018). Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks. *Applied Sciences 2018, Vol. 8, Page 1575*, 8(9), 1575. <https://doi.org/10.3390/APP8091575>
- [5] **Hanbay, K., Talu, M. F., & Özgüven, Ö. F.** (2016). Fabric defect detection systems and methods—A systematic literature review. *Optik*, 127(24), 11960–11973. <https://doi.org/10.1016/j.ijleo.2016.09.110>
- [6] **Uzen, H., Turkoglu, M., & Hanbay, D.** (2021). Texture defect classification with multiple pooling and filter ensemble based on deep neural network. *Expert Systems with Applications*, 175, 114838. <https://doi.org/10.1016/j.eswa.2021.114838>
- [7] **Dong, H., Song, K., He, Y., Xu, J., Yan, Y., & Meng, Q.** (2020). PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection. *IEEE Transactions on Industrial Informatics*, 16(12), 7448–7458. <https://doi.org/10.1109/TII.2019.2958826>
- [8] **Song, G., Song, K., & Yan, Y.** (2020). EDRNet: Encoder-Decoder Residual Network for Salient Object Detection of Strip Steel Surface Defects. *IEEE Transactions on Instrumentation and Measurement*, 69(12), 9709–9719. <https://doi.org/10.1109/TIM.2020.3002277>
- [9] **Xian-guang, F., Xiao-dong, W., Yu-xin, C., & Xin, W.** (2019). Image processing for three defects of topography images by SPM. *Chemometrics and Intelligent Laboratory Systems*, 185, 12–17. <https://doi.org/10.1016/j.chemolab.2018.12.013>
- [10] **Zhang, D., Song, K., Xu, J., He, Y., Niu, M., & Yan, Y.** (2021). MCnet: Multiple Context Information Segmentation Network of No-Service Rail Surface Defects. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–9. <https://doi.org/10.1109/TIM.2020.3040890>
- [11] **Bhatt, P. M., Malhan, R. K., Rajendran, P., Shah, B. C., Thakar, S., Yoon, Y. J., & Gupta, S. K.** (2021, August 1). Image-Based Surface Defect Detection Using Deep Learning: A Review. *Journal of Computing and Information Science in Engineering*. American Society of Mechanical Engineers (ASME). <https://doi.org/10.1115/1.4049535>
- [12] **Yang, Z., Zhu, W., Ma, F., Zhao, J., & Jiang, H.** (2020). Global Context Network for Steel Surface Defect Detection. In *Proceedings of 2020 3rd International*

- Conference on Unmanned Systems, ICUS 2020* (pp. 985–990). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICUS50048.2020.9274836>
- [13] **Luo, Q., Fang, X., Liu, L., Yang, C., & Sun, Y.** (2020). Automated Visual Defect Detection for Flat Steel Surface: A Survey. *IEEE Transactions on Instrumentation and Measurement*, 69(3), 626–644. <https://doi.org/10.1109/TIM.2019.2963555>
- [14] **Zheng, X., Zheng, S., Kong, Y., & Chen, J.** (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. *International Journal of Advanced Manufacturing Technology*, 113(1), 35–58. <https://doi.org/10.1007/s00170-021-06592-8>
- [15] **Yang, T., Zhang, T., & Huang, L.** (2021). Detection of defects in voltage-dependent resistors using stacked-block-based convolutional neural networks. *The Visual Computer*, 37(6), 1559–1567. <https://doi.org/10.1007/s00371-020-01901-w>
- [16] **Ngan, H. Y. T., Pang, G. K. H., & Yung, N. H. C.** (2011). Automated fabric defect detection—A review. *Image and Vision Computing*, 29(7), 442–458. <https://doi.org/10.1016/J.IMAVIS.2011.02.002>
- [17] **Xie, X.** (2008). A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques Figure 1: Example defects on different types of surfaces—from left: Steel. , 7 Xie / *Electronic Letters on Computer Vision and Image Analysis* 1–22
- [18] **Djukic, D., & Spuzic, S.** (2007). Statistical discriminator of surface defects on hot rolled steel. *Proceedings of Image and Vision Computing*, 158–163.
- [19] **Tsai, D. M., Chen, M. C., Li, W. C., & Chiu, W. Y.** (2012). A fast regularity measure for surface defect detection. *Machine Vision and Applications*, 23(5), 869–886. <https://doi.org/10.1007/s00138-011-0403-3>
- [20] **Ojala, T., Pietikäinen, M., & Harwood, D.** (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1), 51–59. [https://doi.org/10.1016/0031-3203\(95\)00067-4](https://doi.org/10.1016/0031-3203(95)00067-4)
- [21] **Wang, Y., Xia, H., Yuan, X., Li, L., & Sun, B.** (2018). Distributed defect recognition on steel surfaces using an improved random forest algorithm with optimal multi-feature-set fusion. *Multimedia Tools and Applications*, 77(13), 16741–16770. <https://doi.org/10.1007/s11042-017-5238-0>
- [22] **Hanbay, K., Golgiyaz, S., & Talu, M. F.** (2017). Real time fabric defect detection system on Matlab and C++/OpenCV platforms. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1–8). Malatya: IEEE. <https://doi.org/10.1109/IDAP.2017.8090180>
- [23] **Ai, Y. hao, & Xu, K.** (2013). Surface Detection of Continuous Casting Slabs Based on Curvelet Transform and Kernel Locality Preserving Projections. *Journal of Iron and Steel Research International*, 20(5), 80–86. [https://doi.org/10.1016/S1006-706X\(13\)60102-8](https://doi.org/10.1016/S1006-706X(13)60102-8)
- [24] **Choi, D. C., Jeon, Y. J., Yun, J. P., & Kim, S. W.** (2011). Pinhole detection in steel slab images using Gabor filter and morphological features. *Applied Optics*, 50(26), 5122–5129. <https://doi.org/10.1364/AO.50.005122>
- [25] **Ghorai, S., Mukherjee, A., Gangadaran, M., & Dutta, P. K.** (2013). Automatic defect detection on hot-rolled flat steel products. *IEEE Transactions on*

- Instrumentation and Measurement*, 62(3), 612–621.
<https://doi.org/10.1109/TIM.2012.2218677>
- [26] **Ren, R., Hung, T., & Tan, K. C.** (2018). A Generic Deep-Learning-Based Approach for Automated Surface Inspection. *IEEE Transactions on Cybernetics*, 48(3), 929–940. <https://doi.org/10.1109/TCYB.2017.2668395>
- [27] **Tabernik, D., Šela, S., Skvarč, J., & Skočaj, D.** (2020). Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 31(3), 759–776. <https://doi.org/10.1007/S10845-019-01476-X/FIGURES/17>
- [28] **Gayubo, F., González, J. L., de La Fuente, E., Miguel, F., & Perán, J. R.** (2006). On-line machine vision system for detect split defects in sheet-metal forming processes. In *Proceedings - International Conference on Pattern Recognition* (Vol. 1, pp. 723–726). <https://doi.org/10.1109/ICPR.2006.902>
- [29] **Yang, J., Li, X., Xu, J., Cao, Y., Zhang, Y., Wang, L., & Jiang, S.** (2018). Development of an optical defect inspection algorithm based on an active contour model for large steel roller surfaces. *Applied Optics*, 57(10), 2490. <https://doi.org/10.1364/ao.57.002490>
- [30] **Yan, H., Paynabar, K., & Shi, J.** (2017). Anomaly Detection in Images With Smooth Background via Smooth-Sparse Decomposition. *Technometrics*, 59(1), 102–114. <https://doi.org/10.1080/00401706.2015.1102764>
- [31] **Sakhare, K., Kulkarni, A., Kumbhakarn, M., & Kare, N.** (2015). Spectral and spatial domain approach for fabric defect detection and classification. In *2015 International Conference on Industrial Instrumentation and Control, ICIC 2015* (pp. 640–644). Pune: Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/IIC.2015.7150820>
- [32] **Yapi, D., Allili, M. S., & Baaziz, N.** (2018). Automatic Fabric Defect Detection Using Learning-Based Local Textural Distributions in the Contourlet Domain. *IEEE Transactions on Automation Science and Engineering*, 15(3), 1014–1026. <https://doi.org/10.1109/TASE.2017.2696748>
- [33] **Liu, G., & Zheng, X.** (2020). Fabric defect detection based on information entropy and frequency domain saliency. *The Visual Computer 2020 37:3*, 37(3), 515–528. <https://doi.org/10.1007/S00371-020-01820-W>
- [34] **Dong, X., Taylor, C. J., & Cootes, T. F.** (2020). A Random Forest-Based Automatic Inspection System for Aerospace Welds in X-Ray Images. *IEEE Transactions on Automation Science and Engineering*. <https://doi.org/10.1109/TASE.2020.3039115>
- [35] **Cao, G., Ruan, S., Peng, Y., Huang, S., & Kwok, N.** (2018). Large-Complex-Surface Defect Detection by Hybrid Gradient Threshold Segmentation and Image Registration. *IEEE Access*, 6, 36235–36246. <https://doi.org/10.1109/ACCESS.2018.2842028>
- [36] **Abid, S.** (2019). Texture defect detection by using polynomial interpolation and multilayer perceptron. *Journal of Engineered Fibers and Fabrics*, 14, 155892501882527. <https://doi.org/10.1177/1558925018825272>
- [37] **Zhang, L., Jing, J., & Zhang, H.** (2015). Fabric Defect Classification Based on LBP and GLCM *. *Journal of Fiber Bioengineering and Informatics*, 8, 81–89. <https://doi.org/10.3993/jfbi03201508>

- [38] **Salem, Y. ben, & Abdelkrim, M. N.** (2020). Texture classification of fabric defects using machine learning. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(4), 4390–4399. <https://doi.org/10.11591/ijece.v10i4.pp4390-4399>
- [39] **Deotale, N. T., & Sarode, T. K.** (2019). Fabric Defect Detection Adopting Combined GLCM, Gabor Wavelet Features and Random Decision Forest. *3D Research*, 10(1), 1–13. <https://doi.org/10.1007/s13319-019-0215-1>
- [40] **Djukic, D., & Spuzic, S.** (2007). Statistical discriminator of surface defects on hot rolled steel. *Proceedings of Image and Vision Computing*, 158–163.
- [41] **Cao, J., Yang, G., & Yang, X.** (2021). A Pixel-Level Segmentation Convolutional Neural Network Based on Deep Feature Fusion for Surface Defect Detection. *IEEE Transactions on Instrumentation and Measurement*, 70. <https://doi.org/10.1109/TIM.2020.3033726>
- [42] **Shu, Y. F., Li, B., Li, X., Xiong, C., Cao, S., & Wen, X. Y.** (2021). Deep learning-based fast recognition of commutator surface defects. *Measurement*, 178, 109324. <https://doi.org/10.1016/J.MEASUREMENT.2021.109324>
- [43] **Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., & Fricout, G.** (2012). Steel defect classification with Max-Pooling Convolutional Neural Networks. *Proceedings of the International Joint Conference on Neural Networks*. <https://doi.org/10.1109/IJCNN.2012.6252468>
- [44] **Weimer, D., Scholz-Reiter, B., & Shpitalni, M.** (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals - Manufacturing Technology*, 65(1), 417–420. <https://doi.org/10.1016/j.cirp.2016.04.072>
- [45] **Jing, J., Ma, H., & Zhang, H.** (2019). Automatic fabric defect detection using a deep convolutional neural network. *Coloration Technology*, 135(3), 213–223. <https://doi.org/10.1111/cote.12394>
- [46] **Li, Y., Zhang, D., & Lee, D. J.** (2019). Automatic fabric defect detection with a wide-and-compact network. *Neurocomputing*, 329, 329–338. <https://doi.org/10.1016/j.neucom.2018.10.070>
- [47] **Mei, S., Wang, Y., & Wen, G.** (2018). Automatic Fabric Defect Detection with a Multi-Scale Convolutional Denoising Autoencoder Network Model. *Sensors*, 18(4), 1064. <https://doi.org/10.3390/s18041064>
- [48] **Yi, L., Li, G., & Jiang, M.** (2017). An End-to-End Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks. *steel research international*, 88(2), 1600068. <https://doi.org/10.1002/SRIN.201600068>
- [49] **Cheon, S., Lee, H., Kim, C. O., & Lee, S. H.** (2019). Convolutional Neural Network for Wafer Surface Defect Classification and the Detection of Unknown Defect Class. *IEEE Transactions on Semiconductor Manufacturing*, 32(2), 163–170. <https://doi.org/10.1109/TSM.2019.2902657>
- [50] **Masci, J., Meier, U., Fricout, G., & Schmidhuber, J.** (2013). Multi-scale pyramidal pooling network for generic steel defect classification. In *Proceedings of the International Joint Conference on Neural Networks*. <https://doi.org/10.1109/IJCNN.2013.6706920>

- [51] **Long, J., Shelhamer, E., & Darrell, T.** (2014). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 640–651. Retrieved Jan. 01, 2021, from <http://arxiv.org/abs/1411.4038>
- [52] **Ren, S., He, K., Girshick, R., & Sun, J.** (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- [53] **Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C.** (2015). SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
- [54] **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2015). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*, 779–788. Retrieved Mar. 16, 2021, from <http://arxiv.org/abs/1506.02640>
- [55] **Qiu, L., Wu, X., & Yu, Z.** (2019). A High-Efficiency Fully Convolutional Networks for Pixel-Wise Surface Defect Detection. *IEEE Access*, 7, 15884–15893. <https://doi.org/10.1109/ACCESS.2019.2894420>
- [56] **Shi, J., Dang, J., Cui, M., Zuo, R., Shimizu, K., Tsunoda, A., & Suzuki, Y.** (2021). Improvement of Damage Segmentation Based on Pixel-Level Data Balance Using VGG-Unet. *Applied Sciences 2021, Vol. 11, Page 518, 11(2)*, 518. <https://doi.org/10.3390/APP11020518>
- [57] **He, Y., Song, K., Meng, Q., & Yan, Y.** (2020). An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. *IEEE Transactions on Instrumentation and Measurement*, 69(4), 1493–1504. <https://doi.org/10.1109/TIM.2019.2915404>
- [58] **Yanan, S., Hui, Z., Li, L., & Hang, Z.** (2019). Rail Surface Defect Detection Method Based on YOLOv3 Deep Learning Networks. *Proceedings 2018 Chinese Automation Congress, CAC 2018*, 1563–1568. <https://doi.org/10.1109/CAC.2018.8623082>
- [59] **Redmon, J., & Farhadi, A.** (2018). YOLOv3: An Incremental Improvement. Retrieved Aug. 09, 2021, from <http://arxiv.org/abs/1804.02767>
- [60] **Yuan, H., Chen, H., Liu, S., Lin, J., & Luo, X.** (2019). A deep convolutional neural network for detection of rail surface defect. *2019 IEEE Vehicle Power and Propulsion Conference, VPPC 2019 - Proceedings*. <https://doi.org/10.1109/VPPC46532.2019.8952236>
- [61] **Li, J., Su, Z., Geng, J., & Yin, Y.** (2018). Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network. *IFAC-PapersOnLine*, 51(21), 76–81. <https://doi.org/10.1016/j.ifacol.2018.09.412>
- [62] **Li, Y., Huang, H., Xie, Q., Yao, L., & Chen, Q.** (2018). Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD. *Applied Sciences*, 8(9), 1678. <https://doi.org/10.3390/app8091678>
- [63] **Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O.** (2018). Autonomous Structural Visual Inspection Using Region-Based Deep Learning for

- Detecting Multiple Damage Types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 731–747. <https://doi.org/10.1111/mice.12334>
- [64] **Chaurasia, A., & Culurciello, E.** (2018). LinkNet: Exploiting encoder representations for efficient semantic segmentation. *2017 IEEE Visual Communications and Image Processing, VCIP 2017, 2018-January*, 1–4. <https://doi.org/10.1109/VCIP.2017.8305148>
- [65] **Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S.** (2016). Feature Pyramid Networks for Object Detection. Retrieved May 05, 2021, from <http://arxiv.org/abs/1612.03144>
- [66] **Seferbekov, S., Iglovikov, V., Buslaev, A., & Shvets, A.** (2018). Feature Pyramid Network for Multi-Class Land Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018-June*, 272–275. <https://doi.org/10.48550/arxiv.1806.03510>
- [67] **Ronneberger, O., Fischer, P., & Brox, T.** (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9351, pp. 234–241). Springer Verlag. https://doi.org/10.1007/978-3-319-24574-4_28
- [68] **Augustauskas, R., & Lipnickas, A.** (2020). Improved Pixel-Level Pavement-Defect Segmentation Using a Deep Autoencoder. *Sensors*, 20(9), 2557. <https://doi.org/10.3390/s20092557>
- [69] **Huang, Y., Qiu, C., & Yuan, K.** (2020). Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1), 85–96. <https://doi.org/10.1007/s00371-018-1588-5>
- [70] **Aslam, Y., Santhi, N., Ramasamy, N., & Ramar, K.** (2020). Localization and segmentation of metal cracks using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 1, 3. <https://doi.org/10.1007/s12652-020-01803-8>
- [71] **Chen, H., Hu, Q., Zhai, B., Chen, H., & Liu, K.** (2020, August 1). A robust weakly supervised learning of deep Conv-Nets for surface defect inspection. *Neural Computing and Applications*. Springer. <https://doi.org/10.1007/s00521-020-04819-5>
- [72] **Defard, T., Setkov, A., Loesch, A., & Audigier, R.** (2021). PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12664 LNCS, 475–489. https://doi.org/10.1007/978-3-030-68799-1_35
- [73] **Rudolph, M., Wandt, B., & Rosenhahn, B.** (2020). Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows. *arXiv*. Retrieved Apr. 22, 2021, from <http://arxiv.org/abs/2008.12577>
- [74] **Wang, J., Xu, G., Li, C., Wang, Z., & Yan, F.** (2021). Surface Defects Detection Using Non-convex Total Variation Regularized RPCA with Kernelization. *IEEE Transactions on Instrumentation and Measurement*, 70. <https://doi.org/10.1109/TIM.2021.3056738>
- [75] **Krizhevsky, A., Sutskever, I., & Hinton, G. E.** (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (pp. 1097–1105). Retrieved from <http://code.google.com/p/cuda-convnet/>

- [76] **Simonyan, K., & Zisserman, A.** (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Information and Software Technology*, 51(4), 769–784. Retrieved Apr. 10, 2020, from <http://arxiv.org/abs/1409.1556>
- [77] **Turkoglu, M., Hanbay, D., Sengur, · Abdulkadir, & Sengur, A.** (2019). Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *Journal of Ambient Intelligence and Humanized Computing*, 1. <https://doi.org/10.1007/s12652-019-01591-w>
- [78] **Türkoğlu, M., & Hanbay, D.** (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering & Computer Sciences*, 23(3), 1636–1651.
- [79] **Turkoglu, M., & Hanbay, D.** (2019). Plant recognition system based on deep features and color-LBP method. In *27th Signal Processing and Communications Applications Conference, SIU 2019*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/SIU.2019.8806592>
- [80] **Firat, H., & Hanbay, D.** (2021). 4CF-Net: Hiperspektral uzaktan algılama görüntülerinin spektral uzamsal sınıflandırılması için yeni 3B evrişimli sinir ağı. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 37(1), 439–454. <https://doi.org/10.17341/GAZIMMFD.901291>
- [81] **Wei, D., Zhou, B., Torralba, A., & T. Freeman, W.** (n.d.). mNeuron: A Matlab Plugin to Visualize Neurons from Deep Models. Retrieved August 18, 2022, from <https://donglaiw.github.io/proj/mneuron/index.html>
- [82] **Tan, M., & Le, Q. v.** (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *36th International Conference on Machine Learning, ICML 2019, 2019-June*, 10691–10700. Retrieved Mar. 04, 2021, from <http://arxiv.org/abs/1905.11946>
- [83] **Wang, J., Lv, P., Wang, H., & Shi, C.** (2021). SAR-U-Net: Squeeze-and-excitation block and atrous spatial pyramid pooling based residual U-Net for automatic liver segmentation in Computed Tomography. *Computer Methods and Programs in Biomedicine*, 208, 106268. <https://doi.org/10.1016/J.CMPB.2021.106268>
- [84] **Üzen, H., Türkoğlu, M., Yanikoglu, B., & Hanbay, D.** (2022). Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects. *Expert Systems with Applications*, 209, 118269. <https://doi.org/10.1016/J.ESWA.2022.118269>
- [85] **Baheti, B., Innani, S., Gajre, S., & Talbar, S.** (2020). Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 358–359). Retrieved from <https://cvit.iiit.ac.in/ncvpr19/>
- [86] **Jing, J., Wang, Z., Rättsch, M., & Zhang, H.** (2020). Mobile-Unet: An efficient convolutional neural network for fabric defect detection. *Textile Research Journal*, 004051752092860. <https://doi.org/10.1177/0040517520928604>
- [87] **Üzen, H., Turkoglu, M., Aslan, M., & Hanbay, D.** (2022). Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection. *The Visual Computer* 2022, 1–20. <https://doi.org/10.1007/S00371-022-02442-0>

- [88] **He, K., Zhang, X., Ren, S., & Sun, J.** (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 2016-Decem, pp. 770–778). IEEE. <https://doi.org/10.1109/CVPR.2016.90>
- [89] **Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S.** (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. <https://doi.org/10.1016/J.NEUCOM.2015.09.116>
- [90] **Lecun, Y., Bengio, Y., & Hinton, G.** (2015). Deep learning. *Nature* *2015* 521:7553, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- [91] **Ari, A., & Hanbay, D.** (2018). Deep learning based brain tumor classification and detection system. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, *26*(5), 2275–2286. <https://doi.org/10.3906/elk-1801-8>
- [92] **Turkoglu, M., & Hanbay, D.** (2019). Recognition of plant leaves: An approach with hybrid features produced by dividing leaf images into two and four parts. *Applied Mathematics and Computation*, *352*, 1–14. <https://doi.org/10.1016/J.AMC.2019.01.054>
- [93] **Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., & Gulyás, B.** (2020, September 2). 3d deep learning on medical images: A review. *Sensors (Switzerland)*. MDPI AG. <https://doi.org/10.3390/s20185097>
- [94] **Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R.** (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.
- [95] **Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H.** (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Retrieved May 06, 2020, from <http://arxiv.org/abs/1704.04861>
- [96] **Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q.** (2016). Densely Connected Convolutional Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 2261–2269. Retrieved from <http://arxiv.org/abs/1608.06993>
- [97] **Firat, H.** (2021). 3B ESA Tabanlı ResNet50 Kullanılarak Hiperspektral Görüntülerin Sınıflandırılması Classification of Hyperspectral Images Using 3D CNN Based ResNet50. *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 6–9. <https://doi.org/10.1109/SIU53274.2021.9477899>
- [98] **Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A.** (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284. <https://doi.org/10.48550/arxiv.1602.07261>
- [99] **Nguyen, L. D., Gao, R., Lin, D., & Lin, Z.** (2019). Biomedical image classification based on a feature concatenation and ensemble of deep CNNs. *Journal of Ambient Intelligence and Humanized Computing*, 1–13. <https://doi.org/10.1007/s12652-019-01276-4>
- [100] **Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A.** (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 07-12-June, pp. 1–9). IEEE. <https://doi.org/10.1109/CVPR.2015.7298594>

- [101] **Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z.** (2015). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*, 2818–2826. Retrieved Aug. 04, 2021, from <https://arxiv.org/abs/1512.00567v3>
- [102] **Firat, H., Asker, M. E., & Hanbay, D.** (2022). Hybrid 3D Convolution and 2D Depthwise Separable Convolution Neural Network for Hyperspectral Image Classification. *Balkan Journal of Electrical and Computer Engineering*, 10(1), 35–46. <https://doi.org/10.17694/BAJECE.1039029>
- [103] **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I.** (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems, 2017-December*, 5999–6009. Retrieved Nov. 27, 2021, from <https://arxiv.org/abs/1706.03762v5>
- [104] **Oktay, O., Schlemper, J., Folgoc, L. le, Lee, M., Heinrich, M., Misawa, K., ... Rueckert, D.** (2018). Attention U-Net: Learning Where to Look for the Pancreas. Retrieved Oct. 20, 2021, from <https://arxiv.org/abs/1804.03999v3>
- [105] **Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E.** (2017). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8), 2011–2023. Retrieved Jun. 18, 2021, from <http://arxiv.org/abs/1709.01507>
- [106] **Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Hounsby, N.** (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Retrieved Oct. 18, 2021, from <https://arxiv.org/abs/2010.11929v2>
- [107] **Roy, A. G., Navab, N., & Wachinger, C.** (2019). Recalibrating Fully Convolutional Networks With Spatial and Channel “Squeeze and Excitation” Blocks. *IEEE Transactions on Medical Imaging*, 38(2), 540–549. <https://doi.org/10.1109/TMI.2018.2867261>
- [108] **Yuxin Li, V., Ostertag, B. J., Ross -, A. E., Li, J., Wang, X., Cui, H., ... Yi-bin, L.** (2021). Fabric Defect Detection Method Based on Improved U-Net. *Journal of Physics: Conference Series*, 1948(1), 012160. <https://doi.org/10.1088/1742-6596/1948/1/012160>
- [109] **Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q.** (2019). ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11531–11539. <https://doi.org/10.48550/arxiv.1910.03151>
- [110] **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K.** (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, 4171–4186. Retrieved Jan. 27, 2022, from <https://arxiv.org/abs/1810.04805v2>
- [111] **Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B.** (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Retrieved Oct. 12, 2021, from <http://arxiv.org/abs/2103.14030>
- [112] **Farahani, M., Gharachorloo, M., Farahani, M., & Manthouri, M.** (2021). ParsBERT: Transformer-based Model for Persian Language Understanding. *Neural*

Processing Letters, 53(6), 3831–3847. <https://doi.org/10.1007/S11063-021-10528-4/TABLES/9>

- [113] **Zhuang, X., Liu, F., Hou, J., Hao, J., & Cai, X.** (2022). Transformer-Based Interactive Multi-Modal Attention Network for Video Sentiment Detection. *Neural Processing Letters*, 1–18. <https://doi.org/10.1007/S11063-021-10713-5/TABLES/8>
- [114] **Yang, S., Deng, Z., Li, X., Zheng, C., Xi, L., Zhuang, J., ... Zhang, Z.** (2021). A novel hybrid model based on STL decomposition and one-dimensional convolutional neural networks with positional encoding for significant wave height forecast. *Renewable Energy*, 173, 531–543. <https://doi.org/10.1016/J.RENENE.2021.04.010>
- [115] **Xu, G., Wu, X., Zhang, X., & He, X.** (2021). LeViT-UNet: Make Faster Encoders with Transformer for Medical Image Segmentation. Retrieved Jan. 06, 2022, from <https://arxiv.org/abs/2107.08623v1>
- [116] **Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... Zhou, Y.** (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. Retrieved Oct. 12, 2021, from <https://arxiv.org/abs/2102.04306v1>
- [117] **Lin, A., Chen, B., Xu, J., Zhang, Z., & Lu, G.** (2021). DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation. Retrieved Oct. 12, 2021, from <https://arxiv.org/abs/2106.06716v1>
- [118] **Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M.** (2021). Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. Retrieved Oct. 12, 2021, from <http://arxiv.org/abs/2105.05537>
- [119] **Qian, C., Zhu, J., Shen, Y., Jiang, Q., & Zhang, Q.** (2022). Deep Transfer Learning in Mechanical Intelligent Fault Diagnosis: Application and Challenge. *Neural Processing Letters*, 1–23. <https://doi.org/10.1007/S11063-021-10719-Z/FIGURES/6>
- [120] **Kaur, T., & Gandhi, T. K.** (2019). Automated Brain Image Classification Based on VGG-16 and Transfer Learning. In *2019 International Conference on Information Technology (ICIT)* (pp. 94–98). India: IEEE. <https://doi.org/10.1109/ICIT48102.2019.00023>
- [121] **Manikonda, S. K. G., & Gaonkar, D. N.** (2019). A Novel Islanding Detection Method Based on Transfer Learning Technique Using VGG16 Network. In *2019 IEEE International Conference on Sustainable Energy Technologies (ICSET)* (pp. 109–114). Bhubaneswar, India: IEEE. <https://doi.org/10.1109/ICSETS.2019.8744778>
- [122] **Swasono, D. I., Tjandrasa, H., & Fathicah, C.** (2019). Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning. In *2019 12th International Conference on Information & Communication Technology and System (ICTS)* (pp. 176–181). Surabaya, Indonesia: IEEE. <https://doi.org/10.1109/ICTS.2019.8850946>
- [123] **François Chollet.** (n.d.). Keras: the Python deep learning API. 2015. Retrieved March 5, 2021, from <https://keras.io/>
- [124] **Badrinarayanan, V., Kendall, A., & Cipolla, R.** (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481–2495. <https://doi.org/10.1109/TPAMI.2016.2644615>

- [125] **Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G., Chen, X., & di Mascio, P.** (2020). Automatic crack detection on road pavements using encoder-decoder architecture. *Materials*, 13(13), 1–18. <https://doi.org/10.3390/ma13132960>
- [126] **Computer Vision Group, F.** (1996). TILDA Textile Texture-Database. Retrieved April 4, 2020, from <https://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html>
- [127] **Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C.** (2019). MVTEC ad-A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 2019-June, pp. 9584–9592). IEEE Computer Society. <https://doi.org/10.1109/CVPR.2019.00982>
- [128] **Zhou, K., Xiao, Y., Yang, J., Cheng, J., Liu, W., Luo, W., ... Gao, S.** (2020). Encoding Structure-Texture Relation with P-Net for Anomaly Detection in Retinal Images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12365 LNCS, 360–377. Retrieved Apr. 23, 2021, from <http://arxiv.org/abs/2008.03632>
- [129] **Tan, D. S., Chen, Y.-C., Chen, T. P.-C., & Chen, W.-C.** (2020). TrustMAE: A Noise-Resilient Defect Classification Framework using Memory-Augmented Auto-Encoders with Trust Regions.
- [130] **Liu, W., Li, R., Zheng, M., Karanam, S., Wu, Z., Bhanu, B., ... Camps, O.** (2019). Towards Visually Explaining Variational Autoencoders. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8639–8648. Retrieved from <http://arxiv.org/abs/1911.07389>
- [131] **Zavrtanik, V., Kristan, M., & Skočaj, D.** (2021). Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112, 107706. <https://doi.org/10.1016/j.patcog.2020.107706>
- [132] **Racki, D., Tomazevic, D., & Skocaj, D.** (2018). A Compact Convolutional Neural Network for Textured Surface Anomaly Detection. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, 2018-January*, 1331–1339. <https://doi.org/10.1109/WACV.2018.00150>
- [133] **Silvestre-Blanes, J., Albero-Albero, T., Miralles, I., Pérez-Llorens, R., & Moreno, J.** (2019). A Public Fabric Database for Defect Detection Methods and Results. *Autex Research Journal*, Vol. 19(4). <https://doi.org/10.2478/aut-2019-0035>
- [134] **Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J.** (2017). Pyramid scene parsing network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 6230–6239. <https://doi.org/10.1109/CVPR.2017.660>
- [135] **Cao, X., Yao, B., Chen, B., & Wang, Y.** (2020). Multi-defect detection for magnetic tile based on SE-U-Net. *ISPCE-CN 2020 - IEEE International Symposium on Product Compliance Engineering-Asia 2020*. <https://doi.org/10.1109/ISPCE-CN51288.2020.9321855>
- [136] **Zhang, Z., Lv, C., Sun, M., & Wang, Z.** (2020). Reliable and Robust Weakly Supervised Attention Networks for Surface Defect Detection. In *2020 7th International Conference on Dependable Systems and Their Applications (DSA)* (pp. 407–414). IEEE. <https://doi.org/10.1109/DSA51864.2020.00071>

- [137] **Huynh, L. D., & Boutry, N.** (2020). A U-Net++ With Pre-Trained EfficientNet Backbone for Segmentation of Diseases and Artifacts in Endoscopy Images and Videos. *undefined*.
- [138] **Natarajan, V., Hung, T. Y., Vaikundam, S., & Chia, L. T.** (2017). Convolutional networks for voting-based anomaly classification in metal surface inspection. In *Proceedings of the IEEE International Conference on Industrial Technology* (pp. 986–991). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICIT.2017.7915495>
- [139] **Wieler, M., & Hahn, T.** (n.d.). Weakly Supervised Learning for Industrial Optical Inspection | Heidelberg Collaboratory for Image Processing (HCI). Retrieved May 7, 2021, from <https://hci.iwr.uni-heidelberg.de/content/weakly-supervised-learning-industrial-optical-inspection>
- [140] **Liu, J., Song, K., Feng, M., Yan, Y., Tu, Z., & Zhu, L.** (2021). Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Optics and Lasers in Engineering*, *136*, 106324. <https://doi.org/10.1016/j.optlaseng.2020.106324>
- [141] **Zhou, Q., Mei, J., Zhang, Q., Wang, S., & Chen, G.** (2021). Semi-supervised fabric defect detection based on image reconstruction and density estimation. *Textile Research Journal*, *91*(9–10), 962–972. <https://doi.org/10.1177/0040517520966733>
- [142] **Yuxiang, W., Shiyi, M., Xiang, X., & Shanshan, H.** (2021). DCSNet: A Surface Defect Classification and Segmentation Model by One-Class Learning. *Journal of Physics: Conference Series*, *1914*(1), 012037. <https://doi.org/10.1088/1742-6596/1914/1/012037>
- [143] **Seçkin, A. Ç., & Seçkin, M.** (2022). Detection of fabric defects with intertwined frame vector feature extraction. *Alexandria Engineering Journal*, *61*(4), 2887–2898. <https://doi.org/10.1016/J.AEJ.2021.08.017>
- [144] **Fan, J., Ling, X., & Liang, J.** (2021). Detection of Surface Defects of Steel Plate Based on ViT. *Journal of Physics: Conference Series*, *2002*(1), 012039. <https://doi.org/10.1088/1742-6596/2002/1/012039>
- [145] Weakly Supervised Learning for Industrial Optical Inspection | Heidelberg Collaboratory for Image Processing (HCI). Retrieved March 5, 2021, from <https://hci.iwr.uni-heidelberg.de/content/weakly-supervised-learning-industrial-optical-inspection>
- [146] **Saralioglu, E., & Gungor, O.** (2020). Semantic segmentation of land cover from high resolution multispectral satellite images by spectral-spatial convolutional neural network. *Geocarto International*. <https://doi.org/10.1080/10106049.2020.1734871>
- [147] **Imani, M., & Ghassemian, H.** (2020). An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges. *Information Fusion*, *59*, 59–83. <https://doi.org/10.1016/J.INFFUS.2020.01.007>
- [148] **Ma, C., Huang, J. bin, Yang, X., & Yang, M. H.** (2019). Robust Visual Tracking via Hierarchical Convolutional Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(11), 2709–2723. <https://doi.org/10.1109/TPAMI.2018.2865311>

- [149] **Chen, H., Pang, Y., Hu, Q., & Liu, K.** (2020). Solar cell surface defect inspection based on multispectral convolutional neural network. *Journal of Intelligent Manufacturing*, 31(2), 453–468. <https://doi.org/10.1007/s10845-018-1458-z>
- [150] **Qiu, Y., Tang, L., Li, B., Niu, S., & Niu, T.** (2020). Uneven Illumination Surface Defects Inspection Based on Saliency Detection and Intrinsic Image Decomposition. *IEEE Access*, 8, 190663–190676. <https://doi.org/10.1109/ACCESS.2020.3032108>
- [151] **Song, L., Lin, W., Yang, Y.-G. G., Zhu, X., Guo, Q., & Xi, J.** (2019). Weak Micro-Scratch Detection Based on Deep Convolutional Neural Network. *IEEE Access*, 7, 27547–27554. <https://doi.org/10.1109/ACCESS.2019.2894863>
- [152] **Li, W., Kazemifar, S., Bai, T., Yan, S., Zheng, W., ... Luo, L.** (2021). An improved Vision Transformer-based method for classifying surface defects in hot-rolled strip steel. *Journal of Physics: Conference Series*, 2082(1), 012016. <https://doi.org/10.1088/1742-6596/2082/1/012016>
- [153] **Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., ... Shao, L.** (2021). Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. Retrieved Jan. 27, 2022, from <https://arxiv.org/abs/2102.12122v2>
- [154] **Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C.** (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. Retrieved Mar. 04, 2021, from <http://arxiv.org/abs/1801.04381>
- [155] **Firat, H., & Hanbay, D.** (2021). Classification of Hyperspectral Images Using 3D CNN Based ResNet50. *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 1–4. <https://doi.org/10.1109/SIU53274.2021.9477899>
- [156] **Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., & Ai, F.** (2020). Training data-efficient image transformers & distillation through attention. Retrieved Dec. 2020, from <https://arxiv.org/abs/2012.12877v2>
- [157] **Liu, Y., Zhuang, C., & Lu, F.** (2021). Unsupervised Two-Stage Anomaly Detection. Retrieved Oct. 14, 2021, from <https://arxiv.org/abs/2103.11671v1>
- [158] **Venkataramanan, S., Peng, K. C., Singh, R. V., & Mahalanobis, A.** (2020). Attention Guided Anomaly Localization in Images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 12362 LNCS, pp. 485–503). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-58520-4_29
- [159] **Cortes, C., Vapnik, V., & Saitta, L.** (1995). Support-vector networks. *Machine Learning* 1995 20:3, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- [160] **Zhang, Y., Li, X., Gao, L., & Li, P.** (2018). A new subset based deep feature learning method for intelligent fault diagnosis of bearing. *Expert Systems with Applications*, 110, 125–142. <https://doi.org/10.1016/J.ESWA.2018.05.032>
- [161] **Turkoglu, M., Yanikoğlu, B., & Hanbay, D.** (2022). PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *Signal, Image and Video Processing*, 16(2), 301–309. <https://doi.org/10.1007/S11760-021-01909-2/TABLES/8>

- [162] **Salem, Y. ben, & Nasri, S.** (2011). Woven Fabric Defects Detection based on Texture classification Algorithm. In *8th International Multi-Conference on Systems, Signals and Devices (SSD 2011)*. Tunisia: Ecole Nationale d'Ingenieurs de Sfax.
- [163] **Jeyaraj, P. R., & Nadar, E. R. S.** (2020). Effective textile quality processing and an accurate inspection system using the advanced deep learning technique. *Textile Research Journal*, 90(9–10), 971–980. <https://doi.org/10.1177/0040517519884124>
- [164] **Jeyaraj, P. R., & Samuel Nadar, E. R.** (2019). Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *International Journal of Clothing Science and Technology*, 31(4), 510–521. <https://doi.org/10.1108/IJCST-11-2018-0135>



ÖZGEÇMİŞ

Ad-Soyad : Hüseyin ÜZEN

ÖĞRENİM DURUMU:

- **Lisans** :2015, Süleyman Demirel Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
- **Yüksek Lisans** :2018, İnönü Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Bölümü
- **Doktora** :2022, İnönü Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Bölümü

MESLEKİ DENEYİM:

- **Üniversite** : Bingöl Üniversitesi, Bilgisayar Mühendisliği, Araştırma Görevlisi, 2017-Devam ediyor

DOKTORA TEZİNDEN TÜRETİLEN ÇALIŞMALAR

1. Uluslararası hakemli dergilerde yapılan çalışmalar (SCI & SSCI):

- Uzen, H., Turkoglu, M., & Hanbay, D. (2021). Texture defect classification with multiple pooling and filter ensemble based on deep neural network. Expert Systems with Applications, 175, 114838.
- Üzen, H., Türkoğlu, M., Yanikoglu, B., & Hanbay, D. (2022). Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects. Expert Systems with Applications, 209, 118269.
- Üzen, H., Turkoglu, M., Aslan, M., & Hanbay, D. (2022). Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection. The Visual Computer, 1-20.
- Üzen, H., Turkoglu, M., & Hanbay, D. (2022). Multi-dimensional Feature Extraction based Deep Encoder-Decoder Network for Automatic Surface Defect Detection, Neural Computing and Applications, 1-20
- Üzen, H., Turkoglu, M., Arı, A., & Hanbay, D. (2023). Piksel seviyesinde yüzey hata tespiti için InceptionV3 tabanlı zenginleştirilmiş öznetelik entegrasyon ağ mimarisi,

Journal of the Faculty of Engineering and Architecture of Gazi University, 2023, 38(2), 721-732

- Üzen, H., Turkoglu, M., Öztürk, D., & Hanbay, D. H. A Novel Hybrid Attention Gate Based on Vision Transformer for the Detection of Surface Defects, Signal, Image and Video Processing, Hakem değerlendirmesinde

2. Ulusal ulakbimde ve dergi parkta taranan dergilerde yapılan çalışmalar:

- Üzen, H., Turkoglu, M., & Hanbay, D. (2021). Result Weighting-Based Resnet Feature Pyramid Network Architecture for Surface Defect Detection. Gazi University Journal of Science Part C: Design and Technology, 9(4), 760-772.
- Üzen, H., Turkoglu, M., Sel, İ., & Hanbay, D. (2021). Derinlemesine Özellik Piramit Ağı Kullanarak Yüzey Hata Tespiti. Computer Science, (Special), 109-115.

3. Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler:

- Üzen, H., Turkoglu, M., Karci, A., & Hanbay, D. (2019, Sep). Automatic Thresholding Method Developed With Entropy For Fabric Defect Detection, in 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), pp. 1–4.
- Üzen, H., Turkoglu, M., & Hanbay, D. (2021, jun). Surface defect detection using deep U-net network architectures”, SIU 2021- 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings, Jun. 2021.