

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ELEKTRİK İLETİM HATLARINDAKİ İZOLATÖR HASARLARININ
DERİN ÖĞRENME (DEEP LEARNING) YÖNTEMİ İLE TESPİTİ**

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Müh. Ahmet TURFANDA

**KASIM 2019
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ELEKTRİK İLETİM HATLARINDAKİ İZOLATÖR HASARLARININ
DERİN ÖĞRENME (DEEP LEARNING) YÖNTEMİ İLE TESPİTİ**

Ahmet TURFANDA

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"ELEKTRİK YÜKSEK MÜHENDİSİ"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 22 / 05 / 2019

Tezin Savunma Tarihi : 01 / 11 / 2019

Tez Danışmanı : Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU

Trabzon 2019

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**Elektrik-Elektronik Mühendisliği Anabilim Dalında
Ahmet TURFANDA Tarafından Hazırlanan**

**ELEKTRİK İLETİM HATLARINDAKİ İZOLATÖR HASARLARININ
DERİN ÖĞRENME (DEEP LEARNING) YÖNTEMİ İLE TESPİTİ**

**başlıklı bu çalışma, Enstitü Yönetim Kurulunun 15 / 10 / 2019 gün ve 1823 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.**

Jüri Üyeleri

Başkan : Prof. Dr. Engin ÖZDEMİR

Üye : Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU

Üye : Dr. Öğr. Üyesi Mehmet ÖZTÜRK



Prof. Dr. Asim KADIOĞLU

Enstitü Müdürü

ÖNSÖZ

Günümüzde, elektrik iletiminde meydana gelebilecek aksaklıklar hayati önem arz etmektedir. Bu aksaklıklardan biri de izolatör arızalarıdır. Elektrik iletim hatlarındaki izolatör hasarlarının tespit edilmesine yönelik birçok yöntem uygulanmıştır. Bunların başlıcaları; akustik (ses dalgaları), VHF (çok yüksek frekanslı) radyo frekans dalgaları, UHF (ultra yüksek frekanslı) radyo frekans dalgaları, mikro dalgalar, manyetik alan dalgaları, kızılötesi tarama, kaçak akım, izolatör direnci ölçümü ve evrışimsel sinir ağı (convolution neural network) ile tespit yöntemleridir.

Yüksek lisans tezi olarak hazırlanan bu çalışmada, izolatör arızalarını tespit etmek ve sınıflandırmak için yukarıda sayılan yöntemlerden evrışimsel sinir ağı (convolution neural network) ile tespit yöntemi tercih edilmiş olup, bu yöntem ek olarak derin öğrenme (deep learning) ve eğitim aktarımı (transfer learning) yöntemleri de bu çalışmada kullanılmıştır. Matlab programı üzerinden “semantic segmentation using deep learning”, “Image Category Classification Using Deep Learning” ve “Transfer Learning Using AlexNet” başlıklı örnek kod blokları, amacımız doğrultusunda değiştirilerek eğitilmiş ağlar elde edilmiştir. Eğitilmiş bu ağlar ile arızalı izolatörler tespit edilmeye çalışılmış ve bu tespitlerdeki başarı oranları incelenmiştir.

Bu tezin hazırlanmasında, şekillenmesinde ve konu seçiminde bana yardımcı olan değerli hocam Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU’na; matlab ve derin öğrenme (deep learning) konularında bilgi ve tecrübelerinden faydalandığım değerli hocam Dr. Öğr. Üyesi Mehmet ÖZTÜRK’e teşekkür ederim. Çalışmam boyunca desteklerini benden esirgemeyen aileme ve yakınlarıma da sonsuz teşekkürler.

Ahmet TURFANDA

Trabzon 2019

TEZ ETİK BEYANNAMESİ

Yüksek lisans tezi olarak sunduğum “Elektrik İletim Hatlarındaki İzolatör Hasarlarının Derin Öğrenme (Deep Learning) Yöntemi ile Tespiti” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU’nun sorumluluğunda tamamladığımı, verileri kendim topladığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 01/11/2019

Ahmet TURFANDA

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	III
TEZ ETİK BEYANNAMESİ	IV
İÇİNDEKİLER	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ	XIV
SEMBOLLER DİZİNİ	XVII
1. GENEL BİLGİLER	1
1.1. Giriş ve Literatür Taraması	1
1.2. Tezin Amacı ve Kapsamı	2
1.3. Elektrik İletim Hatları	3
1.4. İzolatörler	4
1.4.1. Porselen İzolatörler	4
1.4.2. Cam İzolatörler	5
1.5. Derin Öğrenme (Deep Learning) ve Kısa Tarihi	5
2. YAPILAN ÇALIŞMALAR	7
2.1. Fotoğrafların İncelenmesi ve Sınıfların Belirlenmesi	7
2.2. Etiketlerin Oluşturulması ve Fotoğrafların Matlab'a Yüklenmesi	11
2.3. Fotoğrafların Etiketlenmesi ve Etiketlenmiş Fotoğrafların Kaydedilmesi	13
2.4. Etiketlenmiş Fotoğrafların İncelenmesi	15
2.5. Çalışacağımız Matlab Kod Bloğunun Tanıtılması ve Etiketlediğimiz Kırk Resim ile Ağımızın Eğitimi	16
2.6. Etiketlediğimiz Kırk Resim ile Eğitilmiş Ağımızın İzolatör Sınıflarını Bulup Etiketlemesi	30
2.7. Etiketlediğimiz Kırk Resim ile Eğitilmiş Ağımızın İzolatör Sınıflarını Doğru Etiketlemedeki Başarısının On Altı Resimlik “pxdsTest” Resim Kümesi için Ölçümü	34
2.8. Etiketlediğimiz Yüz Resim ile Ağımızın Eğitimi	53

2.9.	Etiketlediğimiz Yüz Resimle Eğitilmiş Ağımızın İzolatör Sınıflarını Doğru Etiketlemedeki Başarısının Kırk Resimlik Test Kümesi İçin Ölçümü	55
2.10.	Etiketlediğimiz Yüz Resimle Eğitilen Ağımızın Etiketlenmemiş Resimlerdeki İzolatör Sınıflarını Tespit Edip Etiketlemesi	69
2.11.	Arızalı ve Sağlam İzolatörleri Tespit Etmedeki Doğruluk Başarım Oranının Yükseltilmesine Yönelik Eğitim Aktarımı (Transfer Learning) Yönteminin Kullanılması	77
2.11.1.	Yüz Yetmiş Adet Resmin İki Sınıflı Etiketlenmesi, Eğitimi ve Eğitim Sonuçları..	78
2.11.2.	Etiketlediğimiz Yüz Yetmiş Resim ile Eğitilmiş Ağımızın Etiketlenmemiş Resimlerdeki İzolatörlerin Yerlerini Tespit Edip Etiketlemesi	86
2.11.3.	Yüz Yetmiş Adet Resimdeki Arızalı ve Sağlam İzolatörlerin "imcrop" Komutu ile Kesilerek Ayrı Klasörlere Kaydedilmesi	90
2.11.4.	Önceden Eğitilmiş Bir Ağ Olan "Alexnet" Ağı Yardımıyla Öğrenme Transferi (Transfer Learning) Yöntemi Kullanılarak İzolatörlerin Arızalı ve Sağlam Olarak Tespiti	91
2.11.5.	Öğrenme Transferi (Transfer Learning) ile Eğitilen Ağımız Yardımıyla Etiketleme İşlemine Tabi Tutulmayan Yeni Resimlerdeki İzolatörlerin Arızalı ve Sağlam Olarak Tespiti	98
3.	SONUÇLAR	101
4.	KAYNAKLAR	105
ÖZGEÇMİŞ		

Yüksek Lisans Tezi

ÖZET

ELEKTRİK İLETİM HATLARINDAKİ İZOLATÖR HASARLARININ DERİN
ÖĞRENME (DEEP LEARNING) YÖNTEMİ İLE TESPİTİ

Ahmet TURFANDA

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU
2019, 106 sayfa

Elektrik enerjisi, önemini sürekli artıran bir ihtiyaç olmakla birlikte, elektriğin son alıcıya ulaştırılması sürecinde elektrik iletiminde kesintilere sebep olan aksaklıkları önceden tespit etmek önem arz etmektedir. Bu aksaklıklardan biri de iletim hatlarındaki izolatörlerin zamanla çeşitli etkenlerle işlevselliğini yitirmesidir. İzolatör arızalarının tespit edilmesinde kullanılan başlıca yöntemler; akustik (ses dalgaları), VHF (çok yüksek frekanslı) radyo frekans dalgaları, UHF (ultra yüksek frekanslı) radyo frekans dalgaları, mikro dalgalar, manyetik alan dalgaları, kızılötesi tarama, kaçak akım, izolatör direnci ölçümü ve evrişimsel sinir ağı (convolution neural network) ile tespittir.

Bu çalışmada, izolatör arızalarını tespit etmeye yönelik evrişimsel sinir ağı ile tespit yöntemi tercih edilmiş olup, bu yönteme ek olarak derin öğrenme (deep learning) ve eğitim aktarımı (transfer learning) yöntemleri de kullanılmıştır.

İletim hatları boyunca çekilmiş yüksek çözünürlüklü fotoğraflardaki izolatörler, arızalı ve sağlam olmak üzere, “matlab image labeler” uygulamasıyla etiketlenmiş, ardından matlab kod bloğu çalıştırılıp eğitilmiş bir ağ oluşturulmuştur. Eğitilmiş bu ağ yardımıyla, etiketlenmiş fotoğraflardaki izolatörlerden arızalı ve sağlam olanlar tespit edilmiştir. Sonrasında, etiketleme yapılmamış yeni fotoğraflardaki arızalı ve sağlam izolatörler, etiketlemeye gerek kalmadan bu eğitilmiş ağ kullanılarak derin öğrenme yöntemiyle tespit edilmiştir. Ardından, eğitim aktarımı yöntemiyle eğitilen ağ kullanılarak arızalı izolatörler tespit edilmiştir. Son olarak, bu tespitlerdeki başarı oranları incelenmiştir.

Anahtar Kelimeler: İzolatör Arızaları, Derin Öğrenme, Eğitim Aktarımı

Master Thesis

SUMMARY

DETERMINATION OF ISOLATOR FAILURES ON THE ELECTRICITY
TRANSMISSION LINES BY DEEP LEARNING METHOD

Ahmet TURFANDA

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Graduate Program
Supervisor: Asst. Prof. Dr. Fatih Mehmet NUROĞLU
2019, 106 Pages

While the need for electricity is increasing over the time, delivering electricity from generation to consumers via transmission lines are important. One of the causes of interruption in the transmission lines is insulator failure due to aging and environmental conditions. The methods used in the detection of defective insulators are; acoustic (sound waves), VHF (very high frequency) radio frequency waves, UHF (ultra high frequency) radio frequency waves, microwaves, magnetic field waves, infrared scanning, leakage current, isolator resistance measurement and convolution neural network.

In this study, detection method using convolution neural network is preferred to detect defective isolator. Deep learning and transfer learning methods are used in addition to this method.

High-resolution photographs containing defective insulators are used for training and testing. The matlab image labeler application has been used to classify types of faults on the classification labeling process. With the help of this trained network, the isolator failure classes in the tagged photographs were determined. Afterwards, the types of isolator failures in the new photographs without labeling were determined by deep learning method with the trained network without the need for labeling. Then, the defective isolators were detected with the help of the network which was trained by transfer learning method. Finally, the success rates in these determinations were examined.

Keywords: Insulator Failures, Deep Learning, Transfer Learning

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.1. Porselen izolatörler	5
Şekil 1.2. Cam izolatörler	5
Şekil 2.1. İletim hattı boyunca çekilmiş fotoğraflardan biri	7
Şekil 2.2. Birinci sınıfımız olan "arizali_cam" etiketini kullandığımız arızalı cam izolatörlerden bazıları	8
Şekil 2.3. İkinci sınıfımız olan "arizali_pors" etiketini kullandığımız arızalı porselen izolatörlerden bazıları	9
Şekil 2.4. Porselen izolatörlerdeki aşınmaya ve kararmaya örnek, (a) Aşınma (üstten aşağı ikinci dilimde), (b) Kararma (en üst dilimde)	9
Şekil 2.5. Üçüncü sınıfımız olan "sağlam_cam" etiketini kullandığımız sağlam cam izolatörlerden bazıları	10
Şekil 2.6. Dördüncü sınıfımız olan "sağlam_pors" etiketini kullandığımız sağlam porselen izolatörlerden bazıları	10
Şekil 2.7. Beşinci sınıfımız olan "belirsiz" etiketini kullandığımız belirsiz porselen izolatörlerin demir direkler arkasında kalmış olanlardan bazıları	11
Şekil 2.8. Beşinci sınıfımız olan "belirsiz" etiketini kullandığımız belirsiz porselen izolatörlerin bir kısmı fotoğraf karesinin dışında kalmış olanlardan bazıları..	11
Şekil 2.9. Matlab /Image labeler ile etiketlerin oluşturulması	12
Şekil 2.10. "Load", "Add images from folder", "Label Pixels", "Label", "Polygon", "Zoom In" butonları	12
Şekil 2.11. Klasördeki ilk fotoğrafımız olan şekil 2.1'deki fotoğrafın etiketlemesinin yapılmış hali	13
Şekil 2.12. Resimde yeşil renkte olan "belirsiz" adlı sınıflandırma etiketlemesinin açıklaması	14
Şekil 2.13. "Export Labels" ve "To File" butonları ile "Export filename" kutucuğu	14
Şekil 2.14. "İmage Labeler" kapatılmak istendiğinde karşımıza çıkan pencere	15
Şekil 2.15. Image labeler uygulamasında, (a) "Import Labels" ve "From File" butonları, (b) "Load" ve "Add images from folder" butonları	15
Şekil 2.16. Kırk resmimizin ilki için, (a) Orijinal resmimiz, (b) Orijinal resmimizin "image labeler" uygulamasıyla etiketlenmiş renkli hali, (c) Orijinal resmimizin "image labeler" uygulamasıyla etiketlenmiş gri tonlamalı hali ...	16
Şekil 2.17. Kırk resmimizin ilk fotoğrafının etiketlenmiş hali "Label_0001.png"	21

Şekil 2.18.	Kırk resmimizin ilk fotoğrafı için "Label_0001.png" etiketli resminin, etiketlenmemiş fotoğrafın üzerine bindirilmiş hali	21
Şekil 2.19.	Etiketlediğimiz kırk resimlik "pxds" resim kümemiz için etiketlerin sahip olduğu piksel sayılarının yüzdelik dağılımlarının grafiksel gösterimi	22
Şekil 2.20.	Çalışma uzayındaki (work space) "pxLayer" değişkeni	26
Şekil 2.21.	Çalışma uzayındaki (work space) "lgraph" değişkeni	26
Şekil 2.22.	RandXReflection (rastgele sol/sağ yansıma), RandXTranslation ve RandYTranslation (rastgele X/Y çevirisi) değerlerinin çalışma uzayı (workspace) çıktısı	28
Şekil 2.23.	"datasource" değişkeninin çalışma uzayındaki (work space) değerleri	29
Şekil 2.24.	"imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için, (a) Eğitim öncesi etiketlediğimiz resim, (b) Eğitim sonrası bilgisayarın etiketlediği resim	32
Şekil 2.25.	Kırk resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş on altı adet resim içeren "imdsTest" resim kümesinin çalışma uzayı çıktısı	35
Şekil 2.26.	Kırk resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş on altı adet resim içeren "pxdsTest" resim kümesinin çalışma uzayı çıktısı	35
Şekil 2.27.	"evaluateSemanticSegmentation" fonksiyonu ile oluşturulan "metrics" değişkeninin çalışma uzayı görünümü	38
Şekil 2.28.	On altı resimlik "imdsTest" resim kümesinin 12'nci resmi (40 resmimizin 13'üncü resmi) için , (a) Eğitim öncesi, etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	44
Şekil 2.29.	On altı resimlik "imdsTest" resim kümesinin 5'inci resmi (40 resmimizin 36'ncı resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	45
Şekil 2.30.	On altı resimlik "imdsTest" resim kümesinin 9'uncu resmi (40 resmimizin 18'inci resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	46
Şekil 2.31.	On altı resimlik "imdsTest" resim kümesinin 15'inci resmi (40 resmimizin 10'uncu resmi) için , (a) Eğitim öncesi, etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	47
Şekil 2.32.	On altı resimlik "imdsTest" resim kümesinin 8'inci resmi (40 resmimizin 4'üncü resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	48
Şekil 2.33.	On altı resimlik "imdsTest" resim kümesinin ilk resmi (40 resmimizin 26'ncı resmi) için "iou" değerlerinin elde edilmesi ve "MeanIoU" değerinin ispatı.....	50

Şekil 2.34.	On altı resimlik “imdsTest” resim kümesinin 11'inci resmi (40 resmimizin 39'uncu resmi) için, (a) Eğitim öncesi, etiketlediğimiz resim, (b) Kırk resimle eğitilmiş ağ kullanılarak bilgisayarın, resimleri etiketlerken baskın sınıfların lehine yatay pozisyondaki porselen izolatörlerin büyük çoğunluğunu sağlam cam izolatör sınıfıyla etiketlemesi	52
Şekil 2.35.	Etiketlediğimiz yüz resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu piksel sayılarının yüzdelik dağılımlarının grafiksel gösterimi	54
Şekil 2.36.	Yüz resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş kırk adet resim içeren “imdsTest” resim kümesinin çalışma uzayı çıktısı	56
Şekil 2.37.	Yüz resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş kırk adet resim içeren “pxdsTest” resim kümesinin çalışma uzayı çıktısı	57
Şekil 2.38.	Kırk resimlik “imdsTest” resim kümesinin 34'üncü resmi (100 resmimizin 13'üncü resmi) için, (a) Eğitim öncesi, etiketlediğimiz resim, (b) Yüz resimle eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	61
Şekil 2.39.	Kırk resimlik “imdsTest” resim kümesinin 28'inci resmi (100 resmimizin 18'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	62
Şekil 2.40.	Kırk resimlik “imdsTest” resim kümesinin 6'ncı resmi (100 resmimizin 31'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	63
Şekil 2.41.	Kırk resimlik “imdsTest” resim kümesinin 21'inci resmi (100 resmimizin 36'ncı resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	64
Şekil 2.42.	Kırk resimlik “imdsTest” resim kümesinin 38'inci resmi (100 resmimizin 57'nci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	65
Şekil 2.43.	Kırk resimlik “imdsTest” resim kümesinin 30'uncu resmi (100 resmimizin 75'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	66
Şekil 2.44.	Kırk resimlik “imdsTest” resim kümesinin 9'uncu resmi (100 resmimizin 61'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	67
Şekil 2.45.	Kırk resimlik “imdsTest” resim kümesinin 24'üncü resmi (100 resmimizin 23'üncü resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	68
Şekil 2.46.	Etiketlenmemiş 100 yeni resimden 48'inci resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	71

Şekil 2.47.	Etiketlenmemiş 100 yeni resimden 96'ncı resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	72
Şekil 2.48.	Etiketlenmemiş 100 yeni resmin 44'üncü resmi için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	73
Şekil 2.49.	Etiketlenmemiş 100 yeni resimden 60'ıncı resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	74
Şekil 2.50.	Etiketlenmemiş 100 yeni resimden 22'nci resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	76
Şekil 2.51.	Klasördeki ilk fotoğrafımız olan şekil 2.1'deki fotoğrafın "image labeler" uygulaması yardımıyla iki sınıflı etiketlemesinin yapılmış hali	78
Şekil 2.52.	Altmış sekiz resimlik "imdsTest" resim kümesinin 45'inci resmi (170 resmimizin 158'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	82
Şekil 2.53.	Altmış sekiz resimlik "imdsTest" resim kümesinin 28'inci resmi (170 resmimizin 19'uncu resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	83
Şekil 2.54.	Altmış sekiz resimlik "imdsTest" resim kümesinin 22'nci resmi (170 resmimizin 87'nci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	84
Şekil 2.55.	Altmış sekiz resimlik "imdsTest" resim kümesinin 26'ncı resmi (170 resmimizin 94'üncü resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	85
Şekil 2.56.	Etiketlenmemiş 70 yeni resmin 67'nci resmi için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	87
Şekil 2.57.	Etiketlenmemiş 70 yeni resmin 21'inci resmi için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	88
Şekil 2.58.	Etiketlenmemiş 70 yeni resimden 14'üncü resim için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim	89
Şekil 2.59.	Kesme işlemi sonucu oluşan arızalı izolatörlerden bazıları	90
Şekil 2.60.	Kesme işlemi sonucu oluşan sağlam izolatörlerden bazıları	91

Şekil 2.61.	“arızali_izolator” ve “sağlam_izolator” sınıf sayılarının eşitlenmesi	92
Şekil 2.62.	Arızalı ve sağlam izolatör klasörleri için, “trainingImages (eğitim kümesini oluşturan resimler)” ile “validationImages (doğrulama kümesini oluşturan resimler)” sayıları	93
Şekil 2.63.	Yüzdelerlik doğruluk başarımlarının sonucunun %72,12 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi	96
Şekil 2.64.	Yüzdelerlik doğruluk başarımlarının sonucunun %66,35 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi	97
Şekil 2.65.	Yüzdelerlik doğruluk başarımlarının sonucunun %64,42 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi	98
Şekil 2.66.	Etiketleme işlemine tabi tutulmayan yeni resimlerimizin içerisindeki izolatörlerden rastgele seçilen 21 adedi için bilgisayarın arızalı ve sağlam izolatör sınıflarını tespiti	100
Şekil 3.1.	On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle; kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "Resim ölçümleri (ImageMetrics)" tablolarındaki “Genel Doğruluk (GlobalAccuracy)” değerlerinin aynı resimler için karşılaştırılması	102

TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 2.1. Resimlerimizin boyutları 4912x7360 iken etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiket piksel sayılarının (PixelCount) ve etiketlerin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi.....	22
Tablo 2.2. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu sınıf etiketleri piksellerinin sıklık yüzdesi	23
Tablo 2.3. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için sınıf ağırlıkları (classWeights) ve frekans (sınıf etiketleri piksellerinin sıklık yüzdesi) karşılaştırması	25
Tablo 2.4. "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için eğitim sonrası bilgisayarın matlab aracılığıyla bulup etiketlediği piksellerin sınıfsal dağılımı	31
Tablo 2.5. On altı resimlik "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için beş sınıfın "iou" değerlerinin komut penceresi çıktısı	33
Tablo 2.6. On altı adet kodlanmış resim içeren "pxdsResult" resim kümesinin ilk resmi (kırk resmimizin 26'ncı resmi) için oluşturulmuş "pixelLabel_01.png" resmi piksellerinin ilk 14 satır ve ilk 21 sütunu için piksel değerleri	37
Tablo 2.7. Resimlerimizin boyutları 360x480 iken on altı resimlik “pxdsTest” resim kümesi için etiketlerimizin piksel sayılarının (PixelCount) ve etiketlerimizin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi.....	39
Tablo 2.8. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu	39
Tablo 2.9. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle eğitilmiş ağ kullanılarak oluşturulan "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)" tablosu	40
Tablo 2.10. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle eğitilmiş ağ kullanılarak oluşturulan veri kümesi ölçümleri (DataSetMetrics) tablosu	41
Tablo 2.11. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu.....	42
Tablo 2.12. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle eğitilmiş ağ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu.....	43

Tablo 2.13. Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.28.b, şekil 2.29.b, şekil 2.30.b, şekil 2.31.b ve şekil 2.32.b'de gösterilenlerinin yüzdesel doğruluk başarımlarını karşılaştırması	49
Tablo 2.14. Resimlerimizin boyutları 4912x7360 iken yüz resimlerimiz için etiket piksel sayılarının (PixelCount) ve etiketlerin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi	53
Tablo 2.15. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu sınıf etiketleri piksellerinin sıklık yüzdesi	54
Tablo 2.16. Etiketlediğimiz yüz resimlik “pxds” resim kümemiz için sınıf ağırlıkları (classWeights) ve frekans (sınıf etiketleri piksellerinin sıklık yüzdesi) karşılaştırması	54
Tablo 2.17. Resimlerimizin boyutları “360x480” iken kırk resimlik “pxdsTest” resim kümesi için etiketlerimizin piksel sayılarının (PixelCount) ve etiketlerimizin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi.....	58
Tablo 2.18. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu	58
Tablo 2.19. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)" tablosu	58
Tablo 2.20. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan veri kümesi ölçümleri (DataSetMetrics) tablosu	59
Tablo 2.21. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu.....	59
Tablo 2.22. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu.....	59
Tablo 2.23. Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.38.b, şekil 2.39.b, şekil 2.40.b, şekil 2.41.b, şekil 2.42.b, şekil 2.43.b, şekil 2.44.b ve şekil 2.45.b'de gösterilenlerinin yüzdesel doğruluk başarımlarını karşılaştırması	69
Tablo 2.24. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu	79
Tablo 2.25. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilen ağ kullanılarak oluşturulan 0-1 aralığına indirgenmiş karışıklık matrisi (normalized confusion matrix) tablosu	79

Tablo 2.26. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "veri kümesi ölçümleri (DataSetMetrics)" tablosu	79
Tablo 2.27. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu	79
Tablo 2.28. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu	80
Tablo 2.29. Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.52.b, şekil 2.53.b, şekil 2.54.b ve şekil 2.55.b'de gösterilenlerinin yüzdesel doğruluk başarımlarını karşılaştırması	86
Tablo 2.30. Kod bloğunun üç kez çalıştırılmasıyla ortaya çıkan üç ayrı sonucun tablo halinde gösterimi	95
Tablo 3.1 On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle ve kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan “Doğruluk” ve “Genel Doğruluk” tablosu.....	102
Tablo 3.2 Eğitim aktarımı (transfer learning) yöntemiyle elde edilen doğruluk başarımlarını sonuçları tablosu	103

SEMBOLLER DİZİNİ

VHF	Very High Frequency (Yüksek Frekans)
UHF	Ultra High Frequency (Ultra Yüksek Frekans)
CNN	Convolution Neural Network (Evrışimsel Sinir Ağı)
RF	Radyo Frekansları (Radio Frequency)
FFT	Fast Fourier Transform (Hızlı Fourier Dönüşümü)
DWT	Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
FEM	Finite Element Method (Sonlu Eleman Metodu)
PDA	Partial Discharge Analysis (Kısmi Deşarj Analizi)
ANN	Artificial Neural Networks (Yapay Sinir Ağları)
HFCT	High Frequency Current Transformer (Yüksek Frekans Akım Dönüşümü)
TEİAŞ	Türkiye Elektrik İletim A.Ş.
TEDAŞ	Türkiye Elektrik Dağıtım A.Ş.
ENH	Enerji Nakil Hatları
GPU	Grafik İşlemci Ünitesi
SegNet	Semantic Segmentation Network (Anlamsal Bölümlendirme Ağı)
RGB	Renkli (Kırmızı Yeşil Mavi tonlamalı) resimler

1. GENEL BİLGİLER

1.1. Giriş ve Literatür Taraması

Günümüzde, elektrik iletiminde yaşanabilecek aksaklıklar birçok konuda olumsuz sonuca neden olmaktadır. Elektrik iletimini kesintiye uğratan birçok etkenden bir tanesi de izolator arızalarıdır. Kilometrelerce uzunluktaki elektrik iletim hatlarındaki sayısız izolatorün hangisinin hasarlı, hangisinin sağlam olduğunun tespit edilmesine yönelik günümüze kadar birçok yöntem ve analiz denenmiştir.

İzolator arızalarının tespit edilmesine yönelik kullanılmış başlıca yöntemler, literatür taramasından anlaşıldığı üzere; akustik (ses dalgaları) [1, 2, 3], VHF (çok yüksek frekanslı) radyo frekans dalgaları [4], UHF (ultra yüksek frekanslı) radyo frekans dalgaları, mikro dalgalar [5], manyetik alan dalgaları [6, 7], kızılötesi tarama, kaçak akım, izolator direnci ölçümü [8] ve evrişimsel sinir ağı (convolution neural network) [9] ile tespittir. En çok tercih edilenleri ise akustik (ses titreşimleri) ve VHF metodudur. Akustik (ses titreşimleri) metodunun temel çalışma mantığı; izolatöre akustik radyatör aracılığıyla ses dalgası çarptırılıp, izolatörden dönen ses dalgalarını akustik vibrometrelerle (titreşim ölçerlerle) almaktır. VHF metodunun çalışma şekli ise; izolatöre RF anten yardımıyla radyo dalgası çarptırılıp, radyo dalgalarını RF anteniyle almaktır. Ardından alınan bu sinyal verileri, çeşitli analiz yöntemleriyle işlenerek, izolatorün arızalı veya sağlam olduğu tespit edilir. Bu çeşitli analiz yöntemleri, daha çok sinyal işleme yöntemlerinden olmakla birlikte, bunlar özetle; dijital sinyal işleme tekniği, cascaded sinyal işleme tekniği, FFT (fast fourier transform) [10], fraktal analizler, ayrık dalgacık dönüşümü (discrete wavelet transform (DWT)) [11], sonlu eleman metodu (finite element method (FEM)) [12, 2, 13], dalgacık enerji katsayıları (wavelet energy coefficients) [14], spektral altbant ağırlık merkezi enerji vektörleri (spectral subband centroid energy vectors) [14], kısmi deşarj analizleri (partial discharge analysis (PDA)) [15], yapay sinir ağları ile akıllı hesaplama (computational intelligence using artificial neural networks (ANN)) [15], yüksek frekans akım dönüşümü ((high frequency current transformer (HFCT)) [16] analiz yöntemlerinden oluşur. Bu sayılan yöntem ve analizlerin her birinin farklı durum ve şartlara bağlı olarak birbirine göre avantaj ve dezavantajları, referans olarak gösterdiğim makalelerde kıyaslanmıştır. Bu kıyaslamalardan dikkate değer olanlardan bir kaçını örnek verelim. VHF band aralığı 30 ile

300 MHz aralığında seçildiğinde daha etkili olduğu görülmüştür. Çünkü 300 MHz üstü olan UHF bandında sinyalin uzun mesafede aşırı sönümleniyor olduğu tespit edilmişken; bu sinyalin 30 MHz frekans değeri altında girişimden (interferens) görece olarak daha çok etkileniyor olduğu tespit edilmiş [4]. VHF band aralığındaki bir sinyalin, UHF band aralığındaki bir sinyale göre SNR (Sinyal Gürültü Oranı) oranının daha düşük olduğu; çevresel girişimden (interferens) etkilenme hassasiyetinin ise daha yüksek olduğu tespit edilmiştir [4]. Diğer bir makalede ise, RF anten aracılığıyla izolator çatlaklarından yansıyan radyo frekansı sinyalleri üç farklı yöntem ile analiz edilmiş. Bu analiz yöntemleri; foruier dönüşümü, istatistiksel analiz (standart sapma-varyans-otokorolasyon katsayısı) ve dalgacık analizidir. Foruier dönüşümünün, diğer iki analize göre, yansıyan sinyallerin ayrık özelliklerini çıkarmada daha başarılı; fakat gürültü ve girişimden (interferens) daha çok etkileniyor olması sebebi ile daha başarısız olduğu tespit edilmiştir. İstatistiksel analizin otokorolasyon açısından, foruier analizinden daha iyi sonuçlar verdiği; dalgacık dönüşümü (zaman-frekans dağılımı) analizinin ise sinyallerin özgün özelliklerini daha net tespit ederek en başarılı analiz olduğu sonucuna varılmış [10].

1.2. Tezin Amacı ve Kapsamı

Bu tez çalışmasının amacı, arızalı izolatorların derin öğrenme (deep learning) yöntemi kullanılarak bilgisayar tarafından tespit edilmesidir. Binlerce fotoğrafın, bilgisayar ekranında insan gözü ile tek tek incelenerek izolator hasarlarının tespit edilmesi gibi zaman alıcı bir uğraşın yerine bilgisayar tarafından arızalı izolatorların otomatik olarak tespit edilmesi amaçlanmıştır.

Bu tez çalışmasında, iletim hattı boyunca çekilmiş izolator fotoğraflarından yeterli sayıda alınıp, bu fotoğraflardaki izolatorlar, matlab programı içerisinde bulunan "image labeler" uygulaması ile; arızalı cam izolator, arızalı porselen izolator, sağlam cam izolator, sağlam porselen izolator olarak etiketlenerek sınıflandırılmış. Daha sonra, matlab programı kütüphanesindeki "semantic segmentation using deep learning (derin öğrenmeyi kullanarak anlamsal bölümlenme)" başlığı altındaki kapsamlı kod blokları, kendi amacımız ve verilerimiz doğrultusunda şekillendirilip değiştirilmiştir. Değiştirdiğimiz kod blokları, matlab programının içinde mevcut olan "deep learning tool box" aracılığıyla çalıştırılmış; bu kod bloğunun çalışması sonucunda matlab programı çalışma uzayı (workspace) içerisinde eğitilmiş bir ağ oluşturulmuştur. Eğitilmiş bu ağ yardımıyla bilgisayarın,

etiketlemeleri yapılmış fotoğraflardaki arıza sınıflarını tespit etmedeki doğruluk başarısı ölçülmüştür. Ardından, yeni fotoğraflar bilgisayara yükledikten sonra artık bunların yeniden etiketlenmesine gerek kalmadan, matlab programında hazırladığımız kod bloğu çalıştırılarak, bu yeni fotoğraflardaki etiketleme işlemi (yani hangi izolatörün arızalı, hangi izolatörün sağlam olduğunun etiketlenerek gösterilmesi), eğittiğimiz bu ağ yardımıyla bilgisayar tarafından otomatik olarak gerçekleştirilmiştir. Derin öğrenme ile eğittiğimiz ağımız sayesinde bilgisayar tarafından, fotoğraflardaki izolatörlerden hangisinin arızalı hangisinin sağlam olduğu %72'lik bir doğruluk başarımla tespit edilmiştir.

1.3. Elektrik İletim Hatları

Elektrik iletim hatları, elektrik üretim tesislerinde üretilen elektrik enerjisinin, bu üretim tesislerden dağıtım hatlarına iletilmesini sağlayan hatlardır. Transformatör istasyonları ile gerilimi düşürülen elektrik dağıtım hatlarına; dağıtım hatları aracılığıyla da çeşitli talepleri olan farklı son kullanıcılara istenilen enerji iletilmiş olur. Elektrik enerjisi, üretim tesislerinden çıkarken yüksek gerilimli olarak şebekeye verilir çünkü iletken hat kayıplarını en aza indirmek için yüksek gerilim ve düşük akım değerleriyle enerjinin uzun hatlar boyunca taşınması bilinen bir prensiptir. Üretim tesislerinden yüksek gerilimli olarak çıkan enerji indirici (gerilim düşürücü) transformatör istasyonlarına vardığında yüksek gerilim değerlerinden orta gerilime seviyelerine düşürülür. Ardından, orta gerilim değerlerini alçak gerilim değerlerine düşüren daha küçük güç kapasitelerine sahip trafo istasyonlarına ulaşan enerji artık son kullanıcılara ulaştırılmak için alçak gerilim seviyelerine düşürülür. Yani özetle, elektrik enerjisi; enerji üretim istasyonları ile yüksek güç kapasiteli trafo istasyonları arasındaki iletim hatlarında yüksek gerilimde; yüksek güç kapasiteli trafo istasyonları ile daha düşük güç kapasiteli trafo istasyonları arasındaki iletim hatlarında orta gerilimde; düşük güç kapasiteli trafo istasyonları ile son kullanıcılar arasındaki iletim hatlarında alçak gerilimde taşınır. Elektrik iletim hatları taşıdıkları gerilim seviyesine göre kabaca dört grupta incelenir:

1. Alçak gerilim iletim hatları (AG hatlar) (Gerilim seviyeleri 1 kV' a kadar).
2. Orta gerilim iletim hatları (OG hatlar) (Gerilim seviyeleri 1 kV ile 35 kV arası).
3. Yüksek gerilim iletim hatları (YG hatlar) (Gerilim seviyeleri 35 kV ile 154 kV arası).
4. Çok yüksek gerilim iletim hatları (Gerilim seviyeleri 154 kV ve daha yukarısı).

Uzun mesafeli yüksek gerilimli enerjinin taşınması havai hatlarla olur. Bu yüksek gerilimli iletim hatlarına enerji nakil hatları (ENH) denir. Enerji nakil hatlarının yer altından gidenleri de vardır fakat bunlar daha çok son kullanıcılara yakın olan düşük gerilimli iletim hatları için tercih edilir. Türkiye’de çok yüksek gerilimli ve yüksek gerilimli (YG) iletim hatlarının kurulumu, bakımı ve diğer sorumlulukları TEİAŞ (Türkiye Elektrik İletim A.Ş.) kurumuna ait iken orta gerilimli (OG) iletim hatları ile alçak gerilimli (AG) iletim hatlarının tesislendirme, onarım ve diğer sorumlulukları TEDAŞ (Türkiye Elektrik Dağıtım A.Ş.) kurumuna aittir; fakat son zamanlarda TEDAŞ Bölge Müdürlüklerinden bir çoğu özelleştirilmiş olup farklı adlarda hizmet vermeye devam etmektedir. TEDAŞ kurumu merkezi denetim yetkisiyle bu özelleştirilmiş kurumları denetlemeye devam etmektedir.

Elektrik iletim hatları nelerden oluşur dersek, kabaca; direkler, iletken teller, izolatörler ve diğer ekipmanlardan oluşur diyebiliriz.

1.4. İzolatörler

İzolatörler, elektrik iletim hatlarındaki iletken telleri direklerden izole etmede kullanılırlar; izolatörlerin bir diğer görevi de iletkenleri taşımaktır. İzolatörler, bu görevleri yerine getirebilmek için yüksek yalıtkanlık değerlerine sahip olmalı ve açık havada kullanıldıklarından olumsuz çevre şartlarına karşı dayanıklı olmalılar. Bu şartları en iyi sağlayan malzemelerden en çok tercih edilenleri porselen ve cam olmuştur. Porselen ve cam izolatörlerden başka silikon izolatörler ve epoksi reçineli izolatörler de vardır fakat malzemelerinin üretimi pahalı olduğu için tercih edilirlikleri düşüktür.

1.4.1. Porselen İzolatörler

Isıya ve fiziksel kuvvetlere karşı dayanıklı olup en çok kullanılan izolatörler porselen izolatörler olmuştur. Üzerleri sır ile kaplanıp kaygan hale getirilir ki kir tutmasın. Yalıtkanlık seviyesi, dielektrik gerilimi cm başına 60 kV civarındadır. Porselenin yalıtkanlığı camdan düşüktür.



Şekil 1.1. Porselen izolatörler

1.4.2. Cam İzolatörler

Cam izolatörlerin dielektrik gerilimi porselen izolatörün yaklaşık iki katı kadardır; bu da yaklaşık olarak 130 kV/cm civarı demek oluyor. Porselen ışığa karşı mat iken cam saydamdır; bu yüzden cam izolatörü, porselen izolatöre göre güneş daha az ısıtır. Cam izolatörler porselen izolatörlere göre daha yenidir ve üretim maliyetleri düşük olduğundan kullanımları her geçen gün artmaktadır.



Şekil 1.2. Cam izolatörler

1.5. Derin Öğrenme (Deep Learning) ve Kısa Tarihi

Derin öğrenme (deep learning) kavramı, yapay sinir ağları konusundaki gelişmelerle birlikte her geçen gün adını daha da duyurmaya devam etmektedir. Derin öğrenme konusunda ilk kayda değer çalışmalar 1950'li yıllardan sonra başlamış fakat ilerlemesi çok yavaş seyretmiştir. Bilgisayar teknolojisinin 1990'lı yıllarından sonra yükselişe geçmesiyle derin öğrenme konusunda yapılan çalışmalar hız kazanmıştır. Yüksek boyutlardaki verinin

işlenmesinin gelişmiş bilgisayar işlemci ünitesi (CPU) sayesinde daha da kolaylaşması sayesinde derin öğrenme algoritmaları ile bu veriler kullanılarak ağ eğitme, eğitilmiş bu ağlar ile verileri sınıflandırma gibi uygulamalar daha da otomatikleşmeye başlamıştır. Her geçen gün verinin daha da yüksek boyutlara çıkması, verilerin işlenmesini daha da zorlaştırmış ve artık merkezi işlem üniteleri (CPU) yetersiz kalmıştır. 2000’li yıllarda grafik işlemci ünitesi (GPU) yani bilgisayar grafik kartlarındaki kayda değer gelişim sayesinde verilerin işlenmesi ve derin öğrenme uygulamaları tekrar yükselişe geçmiştir. Deep learning ile benzer konulardan olan yapay zeka ve makine öğrenmesi günümüzde adından sıkça söz ettirmekte ve geleceği şekillendirmeye devam etmektedir [17].

Derin öğrenme (deep learning) uygulamalarına günümüzden örnekler verecek olursak Apple şirketinin iphone telefonlarındaki Siri uygulaması ve Samsung cep telefonlarındaki Google Asistant uygulamaları konuşulan sesleri metin cümlelerine dönüştürebilirken metin yazılarını da yine konuşma seslerine dönüştürebilmekte, sorulan sorulara mantıklı cevaplar verebilmektedir. Yine derin öğrenme (deep learning), yapay zeka ve makine öğrenmesini içine alan uygulamalardan bir tanesi, bu tez çalışmasının da konusu olan fotoğrafların etiketlenmesi ve sınıflandırılmasıdır.

2. YAPILAN ÇALIŞMALAR

2.1. Fotoğrafların İncelenmesi ve Sınıfların Belirlenmesi

Tez çalışmamızda kullandığımız ve iletim hattı üzerinde bulunan izolatörleri incelememize olanak veren fotoğraflar, yüksek çözünürlüklü olup yaklaşık 10 MB boyutundadır. Bu fotoğrafların bir tanesi örnek olarak Şekil 2.1'de gösterilmiştir.



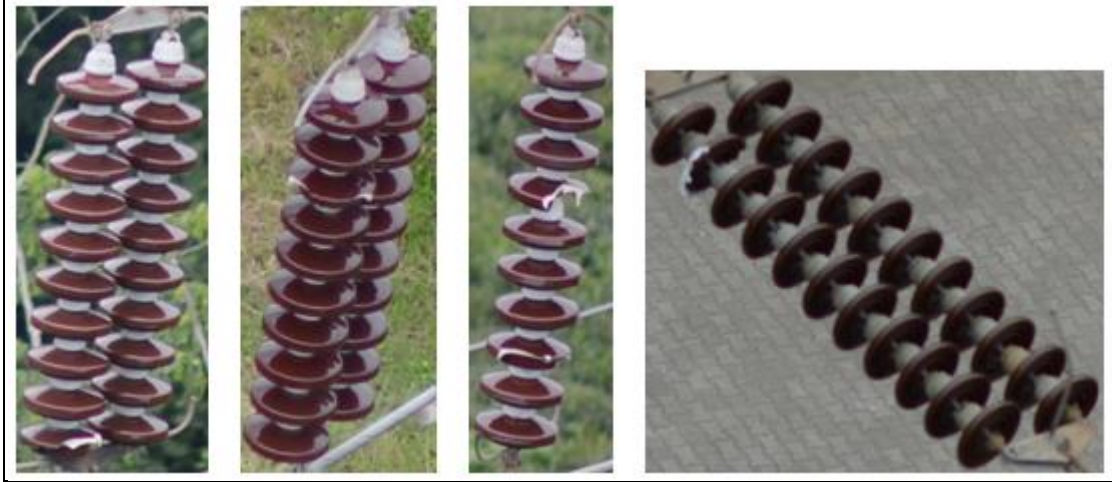
Şekil 2.1. İletim hattı boyunca çekilmiş fotoğraflardan biri

Hat boyunca çekilmiş fotoğraflar incelendiğinde özetle iki çeşit izolatörün kullanıldığı görülmüştür. Bunlar, ağırlıklı olarak cam izolatörler ve daha az kullanılan porselen izolatörlerdir. Cam izolatörlerdeki arızalılık durumu, cam dilimlerin kopmuş olması (dilimlerin kırılarak bütün olarak kopması) (şekil 2.2); porselen izolatör dilimlerindeki arızalılık durumu ise, bütün olarak kopmalar değil de dilimlerde kısmi kırılma (şekil 2.3) şeklinde olması ile özetlenebilir. Bu durumu daha detaylı anlatacak olursak; cam izolatör dilimleri, aşırı gerilimlere dayanamadığında kırılmaya başlar ve kırılan o dilim bütün olarak dökülür yani izolatör grubundan tamamen kopar. Porselen izolatörlerde durum daha farklıdır; porselen dilimlerin aşırı gerilimlerde kırılması, o dilim için tamamen çatlayıp dökülme (yani dilimin tamamen kopması) şeklinde olmaz; bunun yerine kısmi kırılma şeklinde olur; dilimin kısmi kırılmaya uğrayan kısmı izolatör grubundan ayrılırken o dilimin diğer kısmı izolatör grubunda kalmaya devam eder. Cam izolatör dilimlerinin kopması ve porselen dilimlerindeki kısmi kırılmalar, dilimler arası

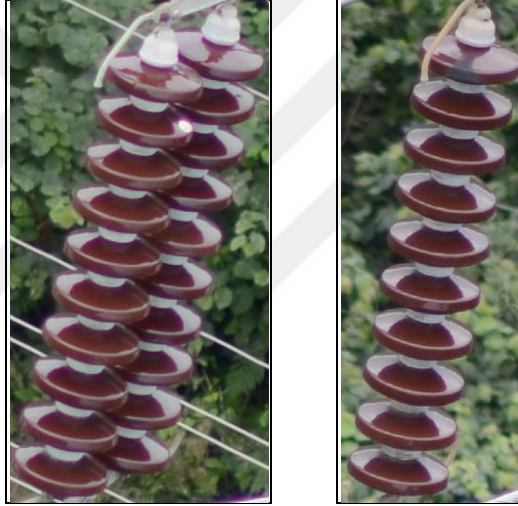
elektrik atlamalarına neden olabileceği için arıza olarak kabul edilmiştir. Bu sayılan arızaların olmadığı cam ve porselen izolatörler ise sağlam olarak kabul edilmiştir (şekil 2.5 ve şekil 2.6). Bazı porselen izolatör dilimlerinde ise kısmi aşınma (şekil 2.4) ve kısmi kararma (şekil 2.4) da görülmüş; bu aşınma ve kararmaların düşük düzeylerde olduğu ve izolatör grubunun işlevini yerine getirmesine engel teşkil etmeyeceği sebebiyle bu aşınma ve kararmalar arıza olarak kabul edilmemiştir. Yine fotoğraflar incelendiğinde, bazı izolatörlerin, demir direklerin arkasında kaldığı görülmüş (şekil 2.7); bu izolatörlerin dilimlerinin net olarak görülemediği ve bu sebeple arızalı mı yoksa sağlam mı olduğu tam tespit edilememiştir. Bazı izolatörler ise bütün olarak fotoğraf karesinin içinde bulunmadığından (şekil 2.8) fotoğraf karesinin dışında kalan izolatör dilimlerinin arızalı olup olmayacağı incelenen o fotoğraf için mümkün olamamıştır. Sayılan bu iki durum içerisine giren cam veya porselen izolatörler ise belirsiz olarak kabul edilmiştir.



Şekil 2.2. Birinci sınıfımız olan "arizali_cam" etiketini kullandığımız arızalı cam izolatörlerden bazıları



Şekil 2.3. İkinci sınıfımız olan "arizali_pors" etiketini kullandığımız arızalı porselen izolatörlerden bazıları



(a)

(b)

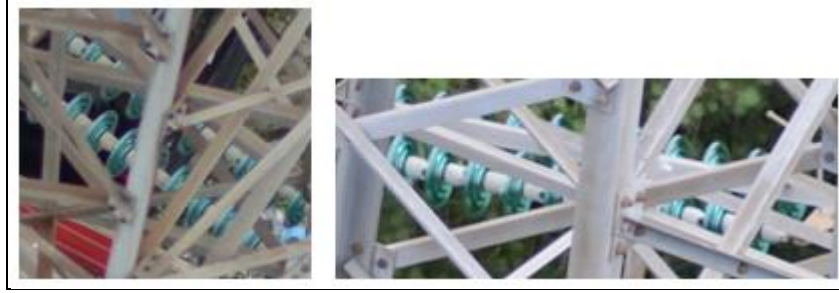
Şekil 2.4. Porselen izolatörlerdeki aşınmaya ve kararmaya örnek, (a) Aşınma (üstten aşağı ikinci dilimde), (b) Kararma (en üst dilimde)



Şekil 2.5. Üçüncü sınıfımız olan "sağlam_cam" etiketini kullandığımız sağlam cam izolatörlerden bazıları



Şekil 2.6. Dördüncü sınıfımız olan "sağlam_pors" etiketini kullandığımız sağlam porselen izolatörlerden bazıları



Şekil 2.7. Beşinci sınıfımız olan "belirsiz" etiketini kullandığımız belirsiz porselen izolatörlerin demir direkler arkasında kalmış olanlardan bazıları

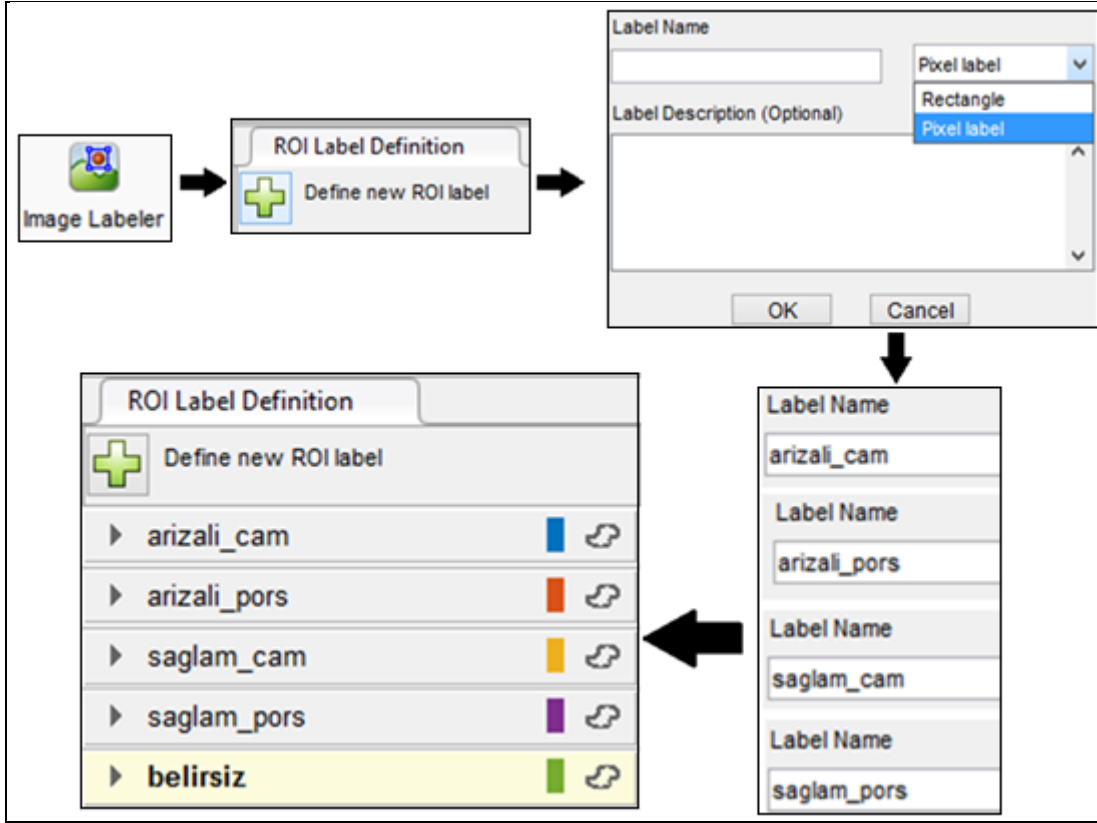


Şekil 2.8. Beşinci sınıfımız olan "belirsiz" etiketini kullandığımız belirsiz porselen izolatörlerin bir kısmı fotoğraf karesinin dışında kalmış olanlardan bazıları

Fotoğraflarımızdaki izolatörleri sınıflandırmak (etiketlemek) için beş sınıf belirlenmiştir. Bu sınıf adları şekil 2.9'da gösterildiği gibi; "arizali_cam", "arizali_pors", "sağlam_cam", "sağlam_pors" ve "belirsiz" olarak belirlenmiş olup bunların uzun adları sırasıyla "arızalı cam izolatör", "arızalı porselen izolatör", "sağlam cam izolatör", "sağlam porselen izolatör" ve "belirsiz izolatör" olarak tanımlanmıştır.

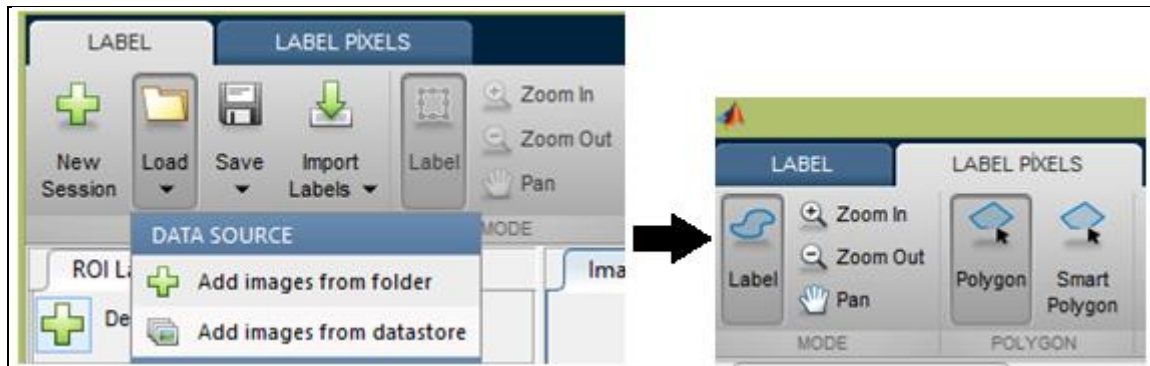
2.2. Etiketlerin Oluşturulması ve Fotoğrafların Matlab'a Yüklenmesi

Fotoğraflarımızdaki izolatörlerin bir önceki başlıkta bahsedilen beş sınıf için etiketlemesi, Matlab 2017b sürümünün Image Labeler uygulaması ile yapılmıştır. Matlab'ın "Applications" sekmesinden Image Labeler uygulaması seçildikten sonra, "Define new ROI label" sekmesine tıklanarak, açılan pencereden karşımıza çıkan iki seçenekten (rectangle or pixel label) "pixel label" seçilip, "label name" kutucuğuna da sınıflandırmada kullanacağımız etiket (label) isimleri sırasıyla yazılmış ve etiketlerimiz oluşturulmuştur. Bu işlemler şekil 2.9'da gösterilmiştir.



Şekil 2.9. Matlab /Image labeler ile etiketlerin oluşturulması

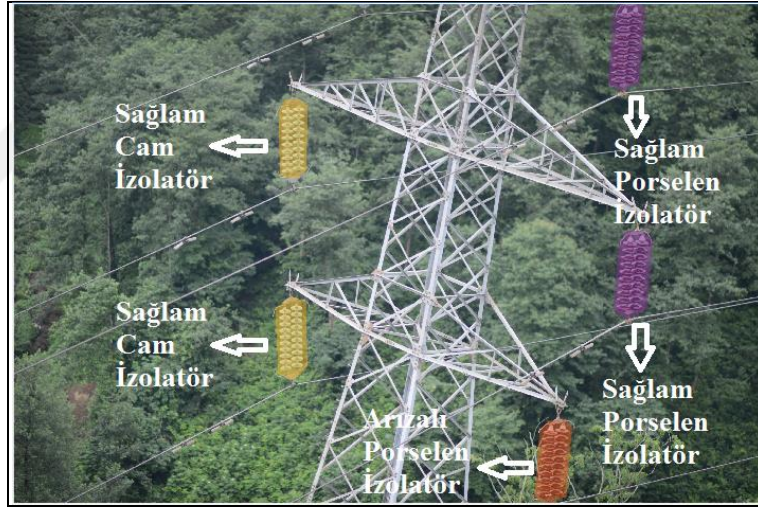
Etiketleme sınıf adlarımız, anlatıldığı gibi belirlendikten sonra "Image Labeler" uygulamasının "Load" butonuna ve ardından "Add images from folder" butonuna tıklanarak fotoğrafları alacağımız klasör seçilip fotoğraflarımız Matlab'a yüklenmiştir. Ardından, "Label Pixels" sekmesine tıklandıktan sonra "Label" ve "Polygon" butonları seçilmiştir (şekil 2.10).



Şekil 2.10. "Load", "Add images from folder", "Label Pixels", "Label", "Polygon", "Zoom In" butonları

2.3. Fotoğrafların Etiketlenmesi ve Etiketlenmiş Fotoğrafların Kaydedilmesi

Fotoğrafta etiketleme yapmak istediğimiz izolatörün, bu beş sınıftan (arizali_cam, arizali_pors, sağlam_cam, sağlam_pors, belirsiz) hangisine girdiğini belirlemek için önce şekil 2.10'da görülen "Zoom In" butonuna tıklanarak fotoğraftaki izolatörlerden hangisinin bu sınıflardan hangisine ait olduğu tespit edildikten sonra etiket isimleri butonlarından (şekil 2.9'da etiket isimleri butonları gösterilmiştir) uygun olanına tıklanılarak mouse ile etiketleyeceğimiz izolatör çevrelenecek şekilde bir etiket çizme işlemi yapılmıştır. Bu işlem belirlenen sayıdaki fotoğrafın tümü için tek tek yapılmıştır. Fotoğraflarımızın bulunduğu klasördeki ilk fotoğrafımız olan şekil 2.1'deki fotoğrafın etiketlemesinin yapılmış hali şekil 2.11'de gösterilmiştir. Sağlam cam, sağlam porselen, arızalı porselen, arızalı cam ve belirsiz izolatörlerin sırasıyla sarı, mor, turuncu, mavi ve yeşil etiketlerle etiketlenmiş hali şekil 2.11 ve şekil 2.12'de gösterilmiştir.

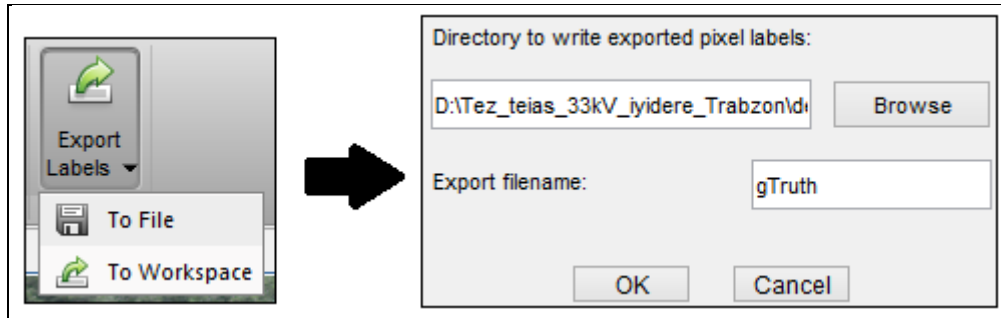


Şekil 2.11. Klasördeki ilk fotoğrafımız olan şekil 2.1'deki fotoğrafın etiketlemesinin yapılmış hali

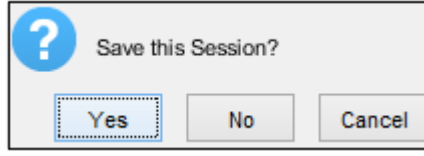


Şekil 2.12. Resimde yeşil renkte olan "belirsiz" adlı sınıflandırma etiketlemesinin açıklaması

Fotoğraflarımızın etiketleme işlemi bittikten sonra Image Labeler Uygulamasındaki "Export Labels" ve ardından "To File" sekmesine basarak "Export filename" kutucuğundan klasör adımız "gTruth" olarak belirlenmiş ve bu "gTruth" adlı klasörün içerisine etiketlenmiş fotoğraflarımız yüklenmiştir (şekil 2.13). Burada dikkat edilmesi gereken bir nokta var. Bu kaydetme (veya yükleme) işlemi yapıldıktan sonra "image labeler" uygulamasını kapatmak istediğimizde karşımıza şekil 2.14'de gösterilen kutucuk çıkar. Burada "yes" butonu seçilmemeli; eğer "yes" butonu seçilirse, daha sonra bu etiketli fotoğrafları "image Labeler" ile açtığımızda etiketlerimizin fotoğraflarda olmadığı görülebilir. Bu durumda yapılan etiketlemelerin tekrar yapılması gerektirir. Bu can sıkıcı sorunla karşılaşmamak için "no" butonuna seçilmelidir.



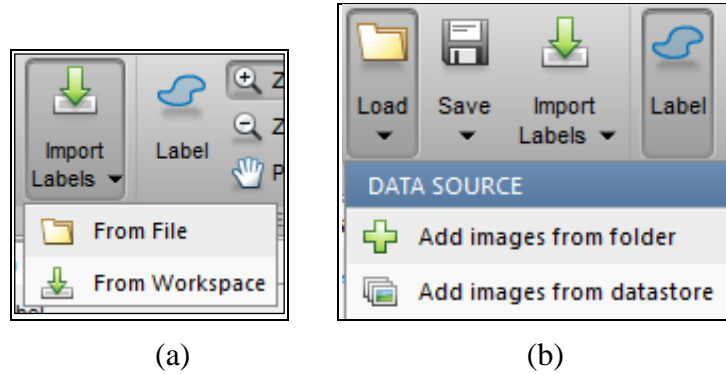
Şekil 2.13. "Export Labels" ve "To File" butonları ile "Export filename" kutucuğu



Şekil 2.14. "İmage Labeler" kapatılmak istendiğinde karşımıza çıkan pencere

2.4. Etiketlenmiş Fotoğrafların İncelenmesi

Etiketlenmiş fotoğraflarımız, oluşturduğumuz klasörün içindeki "gTruth.mat" dosyasının içine kaydedilmiştir. "gTruth.mat" dosyasını açmak için önce "Image Labeler" uygulaması açılır, ardından "Import Labels" butonuna tıklanarak "From File" seçeneği seçilir. Sonra, açılan pencereden etiketli fotoğraflarımızın kayıtlı olduğu "gTruth.mat" dosyası seçilince etiketli fotoğraflarımız açılmış olur (şekil 2.15.a). İlk çalışmamızda 40 resim etiketlenmiştir. Etiketli fotoğraflarımızın sayısını artırmak istersek "Load" ve ardından "Add images from folder" seçilerek (şekil 2.15.b) "gTruth.mat" dosyamıza istediğimiz klasörden yeni fotoğraflar yüklenebilir ve bu yeni fotoğraflar da etiketlenerek eğiteceğimiz ağ için daha fazla etiketlenmiş fotoğraf oluşturulabilir.



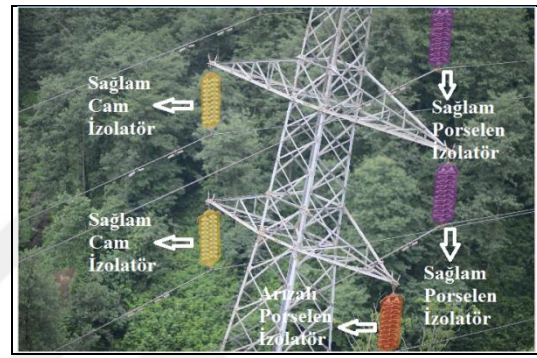
Şekil 2.15. Image labeler uygulamasında, (a) "Import Labels" ve "From File" butonları, (b) "Load" ve "Add images from folder" butonları

"gTruth.mat" dosyasının olduğu klasörde "PixelLabelData" adında bir klasör daha oluşur. Etiketlenmiş fotoğraflar, Bu klasörün içinde ".png" uzantılı olarak otomatik kaydedilir. Örneğin, 40 resimlik klasörümüzün ilk resminin (şekil 2.16.a) "image labeler" uygulamasıyla etiketlenmiş hali olan "Label_0001.png" resmimizi (şekil 2.16.c) inceleyecek olursak, şekil 2.16.a'daki resmin izolatörlerinin olduğu piksellere karşılık

gelen şekil 2.16.c'deki resmin piksellerine baktığımızda; "arızali_pors" etiketli izolatörün olduğu pikseller "Index:2", "saglam_cam" etiketli izolatörün olduğu pikseller "Index:3", "saglam_pors" etiketli izolatörün olduğu pikseller "Index:4" olduğu görülür. Burada hatırlatmamız gerekirse, gri tonlu yani renkli olmayan resimlerde (şekil 2.16.c'deki resmimiz gibi) pikseller minimum "0" maksimum "255" değerini alabilir; "index:0" siyahı, "index:255" ise beyazı temsil eder.



(a)



(b)



(c)

Şekil 2.16. Kırk resmimizin ilki için, (a) Orijinal resmimiz, (b) Orijinal resmimizin "image labeler" uygulamasıyla etiketlenmiş renkli hali, (c) Orijinal resmimizin "image labeler" uygulamasıyla etiketlenmiş gri tonlamalı hali

2.5. Çalışacağımız Matlab Kod Bloğunun Tanıtılması ve Etiketlediğimiz Kırk Resim ile Ağımızın Eğitimi

Çalışacağımız kod bloğu, anlamsal imge segmentasyonu için tasarlanmış bir evrişimsel sinir ağı (CNN - Convolution Neural Network) olan SegNet (Semantic Segmentation Network - Anlamsal Bölümlendirme Ağı) ağını eğitir. Bu kod bloğu, SegNet ağını oluşturmak için VGG-16 ağından (network) başlatılan ağırlıkları (weights) kullanır [18]. VGG-16 ağı, önceden eğitilmiş bir CNN ağı modelidir. Diğer CNN ön eğitilmiş ağı

modellerine bir kaç örnek olarak alexnet, googlenet, importCaffeNetwork ve vgg19 verilebilir. Bu çalışmada kullandığımız kod bloğunun orijinali, Matlab 2017b'nin kütüphanesinde mevcut olan "Semantic Segmentation using Deep Learning" başlıklı çalışmadan alınmış ve kendi amacımız doğrultusunda değiştirilmiştir. Bu kod bloğu çok uzun olduğu için alt kod bloklarına bölünerek anlatılmıştır. Bu başlıkta, kendi resimlerimizle ağımızın nasıl eğitildiği anlatılmıştır. Bundan sonraki başlıkta ise eğitilmiş ağımız yardımıyla matlab programına yükleyeceğimiz yeni fotoğrafların otomatik olarak sınıflara ayrılmasındaki başarı oranı incelenecektir.

```
%Birinci alt kod bloğu (vgg16 ağınaın kurulumu)
```

```
clc, clear, close all;
```

```
vgg16();
```

Birinci alt kod bloğunda, "vgg16()" satırı, vgg16 ağını etkinleştirir. Eğer, vgg16 ağı, bilgisayarda yüklü değilse kod bloğu çalıştırıldığında komut hatası (error) oluşur ve kod bloğu çalışmasını durdurur, "komut penceresinde (command window)" bir sekme (path) oluşur, bu path'e tıklanınca Matlab'ın resmi sitesinden "Neural Network Toolbox Model for VGG-16 Network" ön eğitilmiş CNN ağı, indirilir.

```
%İkinci alt kod bloğu ("images_40resim" klasörünün oluşturulması ve sınıf etiketleri adlarının atanması)
```

```
pretrainedFolder = fullfile(tempdir,'pretraineds','pretrainedSegNet');
```

```
pretrainedSegNet = fullfile(pretrainedFolder,'net.mat');
```

```
outputFolder = fullfile(tempdir,'images_labels');
```

```
imgDir = fullfile(outputFolder,'images_40resim');
```

```
imds = imageDatastore(imgDir);
```

```
I = readimage (imds,1);
```

```
classes = [
```

```
    "arizali_cam"
```

```
    "arizali_pors"
```

```
    "saglam_cam"
```

```
    "saglam_pors"
```

```
    "belirsiz"
```

```
];
```

İkinci alt kod bloğunun ilk iki satırında, eğitim sonunda oluşacak “net” adlı ağıımız için "C:\Users\...\AppData\Local\pretraineds\pretrainedSegNet" konumunda bir klasör oluşturulmuştur. İkinci alt kod bloğunun devam eden satırlarında, "fullfile" komutu ve "tempdir" alt komutu ile bilgisayarımızın "Local" klasörü içindeki "Temp" klasöründe "images_labels" adında bir alt klasör, bu alt klasörün içinde de "images_40resim" adlı klasör oluşturulmuş ve bu "images_40resim" adlı klasörün içine de orjinal fotoğraflarımız (yani etiketlenmemiş halleri) yüklenmiştir. Ardından "imageDatastore" komutu ile "images_40resim" klasörümüz içerisindeki fotoğraflarımız, "imds" değişkenine atanmıştır. Sonrasında, "imds" değişkeni içine atanan fotoğraflarımızdan istenileni ekrana çağırarak için "readimage" komutu kullanılır; "readimage" satırındaki "imds" yanına yazılan sayı kaç ise o sıralamadaki fotoğraf ekrana çağırılmış olur. Ardından "classes" değişkenine izolatör sınıf etiketlerinin adları atanmıştır.

```
% Üçüncü alt kod bloğu (etiket sınıflarımızın değerlerinin ve sıralamasının belirlenmesi)
labelIDs = camvidPixelLabelIDs();
```

Üçüncü alt kod bloğunda, "camvidPixelLabelIDs" fonksiyonunu çağırılır. Bu fonksiyonunun içeriği aşağıda gösterilmiştir.

```
function labelIDs = camvidPixelLabelIDs()
labelIDs = { ...
    [
    1; ... % "arizali_cam"
    ]
    [
    2; ... % "arizali_pors"
    ]
    [
    3; ... % "saglam_cam"
    ]
    [
    4; ... % "saglam_pors"
    ]
}
```

```
[
5; ... % "belirsiz"
]
};
end
```

Önceki başlıklarda bahsedildiği ve şekil 2.16'da de gösterildiği üzere orijinal resimlerimiz "image labeler" uygulamasıyla etiketlendiğinde, bu resimlerin sadece etiketlemesinin yapıldığı izolatör kısımları, beş ayrı sınıf için "Index:1, Index:2, Index:3, Index:4, Index:5" değerlerini alır, resmin geri kalan tüm kısmı (yani etiketlemenin yapılmadığı tüm kısım) ise "Index:0" değerini alarak şekil 2.16.c'de de gösterildiği gibi gri tonlamalı ve ".png" uzantılı olacak şekilde (Label_0001.png, Label_0002.png,... Label_0040.png şeklinde) dönüşüme uğramış olur. İşte, "camvidPixelLabelIDs" fonksiyon bloğundaki 1,2,3,4,5 numaraları da bahsettiğimiz "index" değerlerine karşılık gelen beş ayrı sınıfın temsilidir.

```
%Dördüncü alt kod bloğu (Etiketlenmiş fotoğrafların yükleneceği "labels_40resim"
klasörün oluşturulması)
labelDir = fullfile(outputFolder,'labels_40resim');
```

Dördüncü alt kod bloğunda, "fullfile" komutu ile outputFolder değişkenine atanan "images_labels" adlı klasörümüzün içinde "labels_40resim" adında bir alt klasör oluşturulmuştur.

```
% Beşinci alt kod bloğu ("pixelLabelDatastore" fonksiyonunun oluşturulması ve
"labeloverlay" komutu ile etiketli fotoğrafların etiketsiz fotoğrafların üzerine
yapıştırılması)
pxds = pixelLabelDatastore(labelDir,classes,labelIDs);
C = readimage(pxds,1);
cmap = camvidColorMap;
B = labeloverlay(I,C,'ColorMap',cmap);
figure('Name','B(labels)=labeloverlay(I=(imds,1) , C=(pxds,1)')
imshow(B)
pixelLabelColorbar(cmap,classes);
```

Beşinci alt kod bloğu ilk olarak "pixelLabelDatastore" fonksiyonunu çağırır. "pixelLabelDatastore" fonksiyonu, içerisindeki "labelDir" değişkeni ile "labels" klasörünü; "classes" değişkeni ile beş etiket ismini; labelIDs değişkeni ile "1,2,3,4,5" değerlerinin atandığı izolatör etiket sınıflarını içeren "camvidPixelLabelIDs" fonksiyonunu birlikte işleme sokarak oluşturduğu bilgiyi, açılımı "pixel label data store" yani "piksel etiketi deposu" olan "pxds" değişkenine atar. Ardından, "readimage" komutu ile "pxds" değişkeni içine atanan belirlediğimiz sayıdaki etiketli fotoğrafın ilki (yani label_1.png etiketli fotoğrafı), "C" değişkenine atanır. Sonrasında, "cmap" değişkenine "camvidColorMap" fonksiyonu atanır. Bu fonksiyon aşağıda gösterilmiştir. Bu fonksiyonun yaptığı iş; cmap değişkenine beş etiket ismini atayıp ardından atanan her bir değeri 255'e bölüp, 0-1 aralığına normalize etmektir.

```
function cmap = camvidColorMap()
cmap = [
    128 000 000 % arizali_cam % (128 000 000 = kırmızı)
    128 064 128 % arizali_pors % (128 064 128 = pembe)
    000 000 192 % saglam_cam % (000 000 192 = mavi)
    192 128 128 % saglam_pors % (192 128 128 = somon rengi)
    000 128 192 % belirsiz % (000 128 192 = turkuaz)
];
% Etiket renk değerlerinin 0-1 aralığına normalize edilmesi
cmap = cmap ./ 255;
end
```

Beşinci alt kod bloğundan devam edecek olursak, "labeloverlay" komutu ile I değişkenine atanan ilk resmimizin üzerine, C değişkenine atanan ilk etiketli fotoğrafımız olan "Label_0001.png" resmini oturtur. Üst üste konulmuş bu resmi de B değişkenine atar. Ardından "pixelLabelColorbar" komutu ile B değişkenine atanan resmin üzerinde renk çubuğu (color bar) oluşturulur ve "imshow" komutu ile B resmi ekrana yansır. Yani şekil 2.1'deki resmimizin üzerine şekil 2.17'deki "Label_0001.png" resmi bindirilip şekil 2.18'deki resim elde edilir. Şekil 2.18'da sağdaki colorbar çubuğunda, etiketlerimizin isimleri ve renkleri gösterilmiştir.



Şekil 2.17. Kırk resmimizin ilk fotoğrafının etiketlenmiş hali
“Label_0001.png”



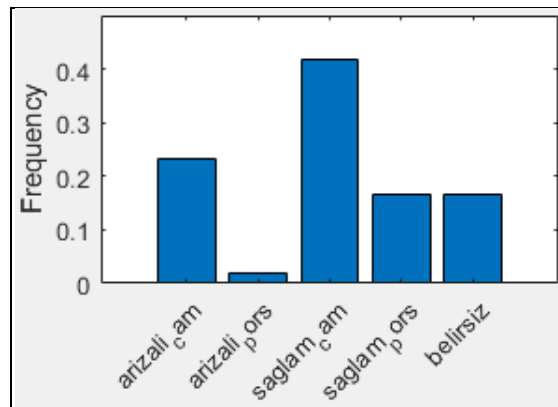
Şekil 2.18. Kırk resmimizin ilk fotoğrafı için “Label_0001.png” etiketli resminin,
etiketlenmemiş fotoğrafın üzerine bindirilmiş hali

```
%Altıncı alt kod bloğu (Kırk resim için sınıf etiketleri başına düşen piksel sayısı analizi)
tbl = countEachLabel(pxds)
frequency = tbl.PixelCount/sum(tbl.PixelCount)
figure('Name','table=classes,frequency')
bar(1:numel(classes),frequency)
xticks(1:numel(classes))
xticklabels(tbl.Name)
xtickangle(45)
ylabel('Frequency')
```

Altıncı alt kod bloğunda; “countEachLabel” komutu, 4912x7360 boyutlarındaki “pxds” veri seti içindeki “.png” uzantılı 40 tane etiketli resim (Label_0001.png, Label_0002.png, ...Label_0040.png) için, her bir etiketin kapladığı toplam piksel sayısını hesaplayarak "PixelCount" sütununda; ilgili etiket sınıfını içeren resimlerin toplam piksel sayısını hesaplayarak "ImagePixelCount" sütununda gösterir. Komut penceresinde (Command window) oluşan “tbl” tablosu, tablo 2.1'de gösterilmiştir. Altıncı alt kod bloğunun ikinci satırında ise etiket piksel sayılarının (PixelCount), etiketlerin bulunduğu resimlerin toplam piksel sayılarına (ImagePixelCount) tablo olarak bölünmesiyle elde edilen değer "frequency" değişkenine atanarak beş etiket için sınıf etiketleri piksellerinin sıklık yüzdesi elde edilerek tablo 2.2 ve şekil 2.19'da gösterilmiştir.

Tablo 2.1. Resimlerimizin boyutları 4912x7360 iken etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiket piksel sayılarının (PixelCount) ve etiketlerin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi

Sınıf (Etiket) Adı	Etiket Piksel Sayıları (PixelCount)	Etiketlerin Bulunduğu Resimlerin Toplam Piksel Sayıları (ImagePixelCount)
'arizali_cam'	11825430	1265331200
'arizali_pors'	1057752	180761600
'saglam_cam'	21252373	1301483520
'saglam_pors'	8446917	723046400
'belirsiz'	8411597	903808000



Şekil 2.19. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu piksel sayılarının yüzdelik dağılımlarının grafiksel gösterimi

Tablo 2.2. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu sınıf etiketleri piksellerinin sıklık yüzdesi

Sınıf (Etiket) Adı	Sınıf Etiketleri Piksellerinin Sıklık Yüzdesi (frequency)
'arizali_cam'	0,2319
'arizali_pors'	0,0207
'saglam_cam'	0,4168
'saglam_pors'	0,1656
'belirsiz'	0,1650

% Yedinci alt kod bloğu (Eğitim için harcanan zamanı ve bellek kullanımını azaltmak için resimlerimizin boyutlarının 4912x7360’tan 360x480 boyutuna küçültülmesi)

```
imageFolder = fullfile(outputFolder,'images_40resimResized',filesep);
imds = resizeCamVidImages(imds,imageFolder);
labelFolder = fullfile(outputFolder,'labels_40resimResized',filesep);
pxds = resizeCamVidPixelLabels(pxds,labelFolder);
```

Yedinci alt kod bloğunda “resizeCamVidImages” fonksiyonu ile “images_40resim” klasörü içindeki resimlerin boyutları 4912x7360’tan 360x480 boyutuna düşürülmüştür. Bunun sebebi eğitim süresinin kısaltılmak istenmesidir çünkü eğitim, bilgisayarın grafik işlemci ünitesinin (GPU) özellikleri düşükse günlerce sürebilir. Bu kod bloğu için bilgisayarın GPU ünitesinin hesaplama kapasitesinin (compute capability) “3.0” ve üzeri olması tavsiye edilmiştir. Bizim çalıştığımız bilgisayarın hesaplama kapasitesinin “2.1” olduğu düşünülürse, “resize” komutunun önemi daha çok anlaşılır. Boyutları küçültülen resimler için “images_40resimResized” klasörü oluşturulmuştur. Yine aynı şekilde, “labels_40resim” klasöründeki “.png” uzantılı etiketli resimlerimiz (Label_0001.png, Label_0002.png,...Label_0040.png) de boyutları 4912x7360’tan 360x480’e düşürülerek “labels_40resimResized” adı ile oluşturulan klasöre konulmuştur.

%Sekizinci alt kod bloğu (Resim sayısının yüzde 60’ı kullanılarak eğitim veri kümesinin, yüzde 40’ı kullanılarak da test veri kümesinin oluşturulması)

```
[imdsTrain,imdsTest,pxdsTrain,pxdsTest] = partitionCamVidData(imds,pxds);
numTrainingImages = numel(imdsTrain.Files)
numTestingImages = numel(imdsTest.Files)
```

Sekizinci alt kod bloğunda, “partitionCamVidData” fonksiyonu, “SegNet” ağıımızın eğitilmesi için, resimlerin rastgele %60’ını seçerek bunları eğitim için kullanır, geri kalan %40’lık kısım ise test için kullanılır. Çalışma uzayında (workspace) oluşan “imdsTrain” ve “pxdsTrain” veri kümeleri, %60’a karşılık gelen 24’er adet resim içerirken “imdsTest” ve “pxdsTest” veri kümeleri ise %40’a karşılık gelen 16’şar adet resim içerir. Kırk resmimiz için “numTrainingImages (eğitim resimleri sayısı)” değişkenine 24; “numTestingImages (test resimleri sayısı)” değişkenine de 16 sayısı atanır.

```
% Dokuzuncu alt kod bloğu (vgg16 ağ katmanlarının (layers) oluşturulması)
```

```
imageSize = [360 480 3];
numClasses = numel(classes);
lgraph = segnetLayers(imageSize,numClasses,'vgg16');
```

Dokuzuncu alt kod bloğunda; “segnetLayers” komutu ile, vgg-16 ağırlıkları (weights) kullanılarak başlatılmış bir “SegNet” ağı oluşturulur. Yani “segnetLayers” komutu otomatik olarak vgg-16 ağırlıklarını “SegNet” ağına transfer eder. Oluşturulan bu “SegNet” ağı, “lgraph” değişkenine atanır. Üstteki kodun ikinci satırında görüldüğü gibi, “SegNet” ağı için resim formatı 360x480 olarak belirlenmiş ve “x3” ile de RGB formatına adapte edilmiştir. Sınıf sayımız da “numClasses” değişkeni ile “segnetLayers” fonksiyonuna atanır.

```
% Onuncu alt kod bloğu (Sınıf ağırlıklarının (class weights) kullanılarak etiket sınıflarının piksellere dağılımının dengelenmesi)
```

```
imageFreq = tbl.PixelCount ./ tbl.ImagePixelCount;
classWeights = median(imageFreq) ./ imageFreq
pxLayer =
pixelClassificationLayer('Name','labels_40resim','ClassNames',tbl.Name,'ClassWeights',classWeights)
```

Tablo 2.3. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için sınıf ağırlıkları (classWeights) ve frekans (sınıf etiketleri piksellerinin sıklık yüzdesi) karşılaştırması.

Sınıf (Etiket) Adı	Sınıf Ağırlıkları (classWeights)	Sınıf Etiketleri Piksellerinin Sıklık Yüzdesi (frequency)
'arizali_cam'	1,0000	0,2319
'arizali_pors'	1,5971	0,0207
'saglam_cam'	0,5723	0,4168
'saglam_pors'	0,8000	0,1656
'belirsiz'	1,0042	0,1650

Onuncu alt kod bloğunun birinci satırında, 40 resim için her bir etiket sınıfının toplam piksel sayısı, “tbl.PixelCount” komutuyla tablo olarak hesaplanır. Yine 40 resim için her bir etiketin bulunduğu resimlerin sınıf ayırt edilmeksizin toplam piksel sayısı da “tbl.ImagePixelCount” komutuyla tablo olarak hesaplanır. Ardından bu iki değer birbirine bölümüyle bulunan değer “imageFreq” değişkenine atanır. Ardından “imageFreq” değişkeninin medyanı, kendisine bölünerek elde edilen değer “classWeights” değişkenine atanır; bu değer sınıf ağırlıkları (class weights) olarak adlandırılır. Sınıf ağırlıkları kullanılarak sınıflar daha dengeli hale getirilir. Bu, ağın dengesizliğini gidermek için yapılır çünkü öğrenme (eğitim) baskın sınıfların lehine ön yargılıdır. Tablo 2.3'te de görüldüğü üzere “saglam_cam” etiketimizin piksel yüzdesi %41,68 iken “saglam_pors” etiketimizinki %16,50'dir. Bu durumda, sınıf ağırlıkları kullanılarak sınıflar daha dengeli hale getirilmeseydi, bu dengesizlik öğrenme sürecine zarar verebilirdi yani eğitim sonucunda oluşan eğitilmiş ağımlar yeni fotoğraflarda olması gerekenden daha fazla sayıda pikseli “saglam_cam” etiketiyle etiketleyecek ve olması gerekenden daha az sayıda pikseli de “saglam_pors” etiketiyle etiketleyecekti. Bu durumda “saglam_cam” etiketi zamanla tüm resmi kaplamaya başlayabilirken “saglam_pors” etiketi zamanla yok olmaya doğru giden bir sürece girecekti. Bu sorun sınıf ağırlıkları (class weights) kullanılarak giderilmiş olur. Tablo 2.3'te dikkat edilirse " sınıf etiketleri piksellerinin sıklık yüzdesi (frequency)" ile "sınıf ağırlıkları (classWeights)" ters orantılıdır. Yani sınıf etiketi piksel yüzdesi düşük olan etiketin sınıf ağırlığı yükseltilmişken yüksek olanın ise sınıf ağırlığı düşürülmüştür. Ardından, “pixelClassificationLayer” fonksiyonu ile “sınıf ağırlıklarımız (class weights)” ve “.png” uzantılı etiketli resimlerimizin olduğu “labels” klasörümüz, “pxLayer” değişkenine atanır (şekil 2.20).

Property	Value
Name	'labels'
ClassNames	5x1 cell
ClassWeights	[1;1.5971;0.5723;0.8000;1.0042]
OutputSize	'auto'
LossFunction	'crossentropyex'

Şekil 2.20. Çalışma uzayındaki (work space) “pxLayer” değişkeni

```
% On birinci alt kod bloğu (vgg16 ağ katmanlarına (layers) etiketli resimlerimizin
bulunduğu "labels" klasörümüzün tanıtılması )
lgraph = removeLayers(lgraph,'pixelLabels');
lgraph = addLayers(lgraph, pxLayer);
lgraph = connectLayers(lgraph,'softmax','labels_40resim');
```

On birinci alt kod bloğunda, “pixelLabels” olan geçerli “pixelClassificationLayer” ögesi “removeLayers” komutu ile kaldırılır. Ardından, “SegNet” ağına “addLayers” komutu ile “pxLayer” katmanı eklenir. Yeni eklenen bu katman “connectLayers” komutu kullanılarak ağın geri kalanına bağlanır. Böylece "lgraph.Connections" matrisinin son satırındaki "pixelLabels" kaldırılıp yerine bizim ".png" uzantılı etiketli resimlerimizi içeren "labels_40resim" klasörümüz yerleştirilmiş olur (şekil 2.21).

	1	2
	Source	Destination
99	'decoder1_relu_1'	'softmax'
100	'softmax'	'pixelLabels'

	1	2
	Source	Destination
99	'decoder1_relu_1'	'softmax'
100	'softmax'	'labels'

Şekil 2.21. Çalışma uzayındaki (work space) “lgraph” değişkeni

```
% On ikinci alt kod bloğu (Eğitim seçeneklerinin belirlenmesi)
```

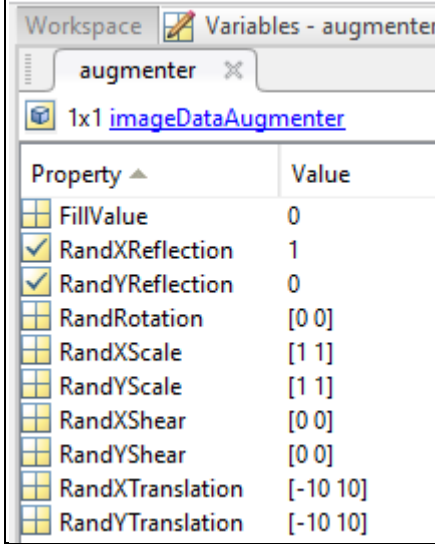
```
options = trainingOptions('sgdm', ...
    'Momentum',0.9, ...
    'InitialLearnRate',1e-3, ...
    'L2Regularization',0.0005, ...
    'MaxEpochs',100, ...
    'MiniBatchSize',3, ...
    'Shuffle','every-epoch', ...
    'CheckpointPath', tempdir, ...
    'VerboseFrequency',2);
```

On ikinci alt kod bloğu eğitimde kullanacağımız seçenekleri bize sunar. Burada “MiniBatchSize” değeri, bilgisayarımızın sahip olduğu GPU bellek kapasitesine bağlı olarak artırılabilir veya azaltılabilir. GPU belleği düşük ise “MiniBatchSize” değeri “1” değerine kadar düşürülebilir fakat değeri düşürdükçe eğitimin süresinin uzayacağı unutulmamalı. Bu çalışmada kullandığımız bilgisayarımızın GPU bellek kapasitesine en uygun olan “MiniBatchSize” değerinin "3" olduğuna karar verilmiş ve eğitime bu değer ile girilmiştir.

```
% On üçüncü alt kod bloğu (Veri Büyütme (Data Augmentation))
```

```
augmenter = imageDataAugmenter('RandXReflection',true,...
    'RandXTranslation',[-10 10],'RandYTranslation',[-10 10]);
```

On üçüncü alt kod bloğunda, “imageDataAugmenter” komutu ile veri büyütme (data augmentation) işlemi yapılmış ve bu işlem “augmenter” değişkenine atanmıştır. Veri büyütme işlemi, eğitim sırasında ağı daha fazla örnek sağlamak için kullanılır böylece ağın doğruluğunu artırmaya yardımcı olur. Burada, veri büyütme için “rastgele sol/sağ yansıma (RandXReflection)” ve “rastgele X/Y çevirisi (RandXTranslation ve RandYTranslation)”, +/- 10 piksel hassasiyet değeriyle kullanılmıştır (şekil 2.22).

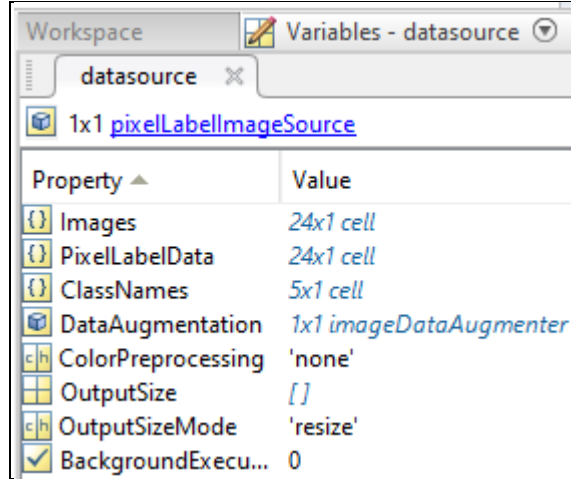


Property	Value
FillValue	0
<input checked="" type="checkbox"/> RandXReflection	1
<input checked="" type="checkbox"/> RandYReflection	0
RandRotation	[0 0]
RandXScale	[1 1]
RandYScale	[1 1]
RandXShear	[0 0]
RandYShear	[0 0]
RandXTranslation	[-10 10]
RandYTranslation	[-10 10]

Şekil 2.22. RandXReflection (rastgele sol/sağ yansıma), RandXTranslation ve RandYTranslation (rastgele X/Y çevirisi) değerlerinin çalışma uzayı (workspace) çıktısı

```
% On dördüncü alt kod bloğu (Eğitimin başlatılması)
datasource = pixelLabelImageSource(imdsTrain,pxdsTrain, ...
    'DataAugmentation',augmenter);
doTraining = true;
if doTraining
    [net, info] = trainNetwork(datasource,lgraph,options);
else
    data = load(pretrainedSegNet);
    net = data.net;
end
```

On dördüncü alt kod bloğunda, “pixelLabelImageSource” komutu yardımıyla “imdsTrain”, “pxdsTrain” ve “augmenter” değişkenleri birlikte işleme sokularak oluşturulan bilgi, “datasource” değişkenine atanır. Burada “imdsTrain”, ağı eğitmek için 40 tane resmimizden bilgisayarın rastgele seçtiği 24 resmi ($40 \times 60 = 24$) temsil ederken; pxdsTrain” ise aynı ağı eğitmek için 40 tane “.png” uzantılı etiketli resimlerimizden bilgisayarın rastgele seçtiği yine 24 resmi ($40 \times 60 = 24$) temsil eder. “augmenter” değişkeni şekil 2.22’de, “datasource” değişkeni ise şekil 2.23’te gösterilmiştir.



Property	Value
Images	24x1 cell
PixelLabelData	24x1 cell
ClassNames	5x1 cell
DataAugmentation	1x1 imageDataAugmenter
ColorPreprocessing	'none'
OutputSize	[]
OutputSizeMode	'resize'
BackgroundExecu...	0

Şekil 2.23. “datasource” değişkeninin çalışma uzayındaki (work space) değerleri

On dördüncü alt kod bloğumuzun "dotraining" satırında, "dotraining=true" veya "dotraining=false" seçimi yapılmalıdır. Eğer “dotraining=false” yazarak kodu çalıştırsak, kod bloğundaki "else" ile "end" arasındaki komutlar çalışarak “SegNet” ağıımız eğitime girmeden “load” komutu ile “data” değişkenine “pretrainedSegNet” adlı önceden eğitilmiş ağıın bilgisini aktarır, ardından “data.net” komutu da bu bilgiyi “net” adlı değişkene aktarır. Böylece ağıımız eğitime girmeyecek, bunu yerine önceden eğitilmiş “pretrainedSegNet” ağını kullanarak ilerlemeye devam edecektir. Fakat “dotraining=true” yazarak kodu çalıştırsak, "if" ile "else" arasındaki komutlar çalışarak “trainNetwork” fonksiyonu; “datasource” ,”lgraph” ve “options” verilerini kendi özel yazılımında işleme sokarak “SegNet” ağıımızı eğitmeye başlar. Komut satırına "dotraining=true" yazılarak ağıımızın eğitilmesine başlanır.

Bu eğitim, bilgisayarın GPU bellek kapasitesine bağlı olarak uzun veya kısa sürebilir. Bu kod bloğunu çalıştıracak bilgisayar için tavsiye edilen hesaplama kapasitesi (compute capability) “3.0” ve üzeri iken bizim çalıştığımız bilgisayarın hesaplama kapasitesi 2.1'dir. Bilgisayarımızın GPU modeli, GeForce GT 630 olduğu için (düşük bellek kapasiteli grafik işlemci ünitesi) etiketlediğimiz 40 resim ile ağıın eğitilmesi, "39366,61" saniye yani yaklaşık 11 saat sürmüştür.

2.6. Etiketlediğimiz Kırk Resim ile Eğitilmiş Ağımızın İzolatör Sınıflarını Bulup Etiketlemesi

```
% On beşinci alt kod bloğu (Etiketlediğimiz kırk resim ile eğitilmiş ağımız yardımıyla bilgisayarın izolatör etiket sınıflarını bulması)
I2 = read(imdsTest);
figure('Name','I2=(imdsTest)')
imshow(I2)
C2 = semanticseg(I2, net);
B2 = labeloverlay(I2, C2, 'Colormap', cmap, 'Transparency',0.4);
figure('Name','B2=labeloverlay(I2, C2)')
imshow(B2)
pixelLabelColorbar(cmap, classes);
```

On beşinci alt kod bloğunda, eğitilmiş ağımızın resimlerimizdeki izolatör sınıflarını ne derece belirleyebildiğini test edebilmek için "semanticseg" komutu kullanılmıştır. Öncesinde örnek olarak 40 resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş 16 adet resim içeren "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi), "read(imdsTest)" komutuyla ("readimage(imdsTest,1)" komutu da aynı işi yapar) "I2" değişkenine atanır. Ardından "semanticseg" komutumuz da "net" adlı eğitilmiş ağımızı kullanarak bu "I2" değişkenine atanmış resmin tüm piksellerini sınıf etiketleriyle kodlar. Böylece tüm pikselleri sınıf etiketleriyle kodlanmış bu resim "C2" değişkenine atanır. 360x480 boyutlarındaki (360 satır ve 480 sütun) "C2" değişkenimizin çalışma uzayındaki (work space) görünümün küçük bir parçası (ilk 11 satırı ve ilk 5 sütunu), tablo 2.4'te gösterilmiştir.

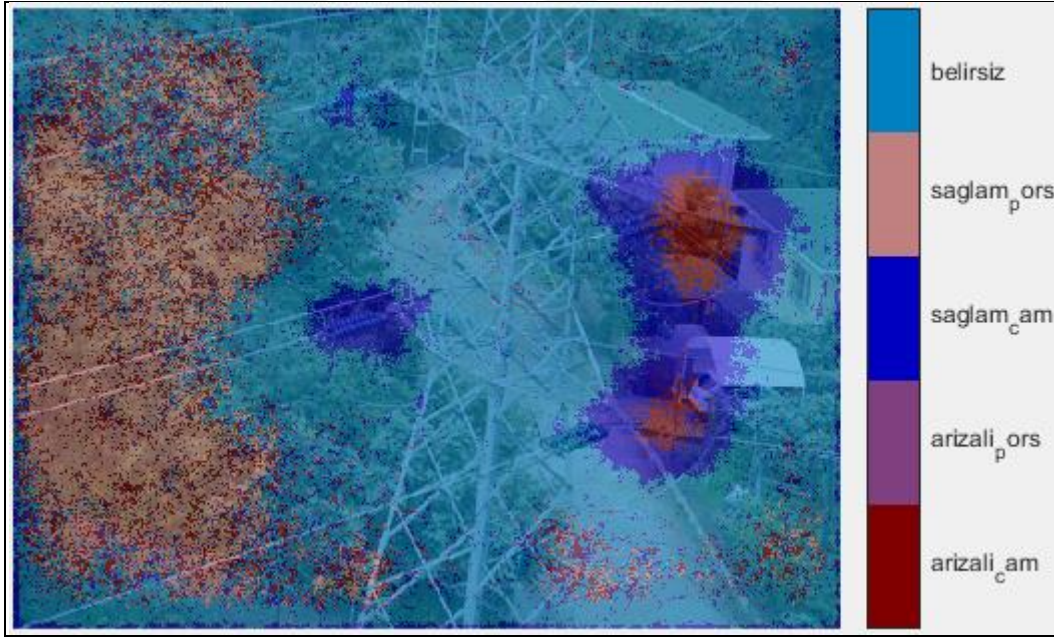
Tablo 2.4. "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için eğitim sonrası bilgisayarın matlab aracılığıyla bulup etiketlediği piksellerin sınıfsal dağılımı

	1	2	3	4	5
1	'belirsiz'	'saglam_cam'	'belirsiz'	'belirsiz'	'belirsiz'
2	'saglam_cam'	'saglam_cam'	'belirsiz'	'belirsiz'	'belirsiz'
3	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'
4	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'
5	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'
6	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'
7	'belirsiz'	'belirsiz'	'belirsiz'	'saglam_cam'	'belirsiz'
8	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'saglam_cam'
9	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'saglam_cam'
10	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'
11	'saglam_cam'	'belirsiz'	'belirsiz'	'belirsiz'	'belirsiz'

On beşinci alt kod bloğundaki "labeloverlay" komutu ise "imdsTest" resim klasörümüzün ilk resminin (I2) üzerine sınıf etiketleriyle otomatik olarak kodlanmış resmimizi bindirir ve üst üste bindirilmiş bu resmi "B2" değişkenine atar (şekil 2.24.b).



(a)



(b)

Şekil 2.24. "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için, (a) Eğitim öncesi etiketlediğimiz resim, (b) Eğitim sonrası bilgisayarın etiketlediği resim

```
% On altıncı alt kod bloğu (Etiketlediğimiz kırk resim ile eğitilmiş ağımız yardımıyla
bilgisayarın izolatör etiket sınıflarını doğru bulmadaki başarı oranının "jaccard endeksi" ve
"iou" değerleriyle ölçülmesi)
```

```
expectedResult = read(pxdsTest);
```

```
iou = jaccard(C2,expectedResult);
```

```
table(classes,iou)
```

On altıncı alt kod bloğunda, şekil 2.24.a'da gösterilen kendi çizdiğimiz etiketler (expectedResult) ile şekil 2.24.b'de gösterilen bilgisayarın bulunduğu etiketler (C2), üst üste bindirilip kesişimleri alınır; böylece bizim çizdiğimiz etiketler ile bilgisayarın bulunduğu etiketlerin örtüştüğü (üst üste geldiği) piksellerin sayısının (yani kesişimlerinin), bizim çizdiğimiz etiketler ile bilgisayarın bulunduğu etiketleri kapsayan piksellerinin birleşim kümesine oranı bize "jaccard" endeksi olarak da bilinen "iou" değerini (intersection over union) yani "birleşimlerinin kesişimini" verir. Bir örnekle açıklayalım. Örneğin sağlam cam izolatör etiketi olan "saglam_cam" etiketini baz alalım. Şekil 2.24.a'daki mavi etiketli pikseller ile şekil 2.24.b'deki mavi etiketli piksellerin kesiştiği yani üst üste örtüştüğü piksel sayısının, şekil 2.24.b'deki mavi etiketli piksellerin sayısına yüzdesel oranı bize "saglam_cam" etiketinin iou değerini (intersection over union) verir. On altıncı alt kod bloğunda, "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için hesapladığımız "iou" değerinin komut penceresi çıktısı tablo 2.5'te gösterilmiştir.

Tablo 2.5. On altı resimlik "imdsTest" resim klasörümüzün ilk resmi (40 resmimizin 26'ncı resmi) için beş sınıfın "iou" değerlerinin komut penceresi çıktısı

Sınıf (Etiket) Adı (classes)	iou (intersection over union) (birleşimlerinin kesişimi) değeri
'arizali_cam'	0,0305
'arizali_pors'	0,0000
'saglam_cam'	0,0375
'saglam_pors'	0,0000
'belirsiz'	0,0104

Tablo 2.5 incelendiğinde, "arizali_pors" ve "saglam_pors" sınıflarının "iou" değerlerinin sıfır olduğu görülür. Bunun sebebi, çalışılan bu resmin bu iki sınıfı

içermemesidir. "arizali_cam", "saglam_cam" ve "belirsiz" etiketlerinin ise "iou" değerleri sırasıyla %3,045, %3,75 ve %1,04'tür. Bu üç değerin düşük çıkma sebebi ise bilgisayarın, eğitilmiş ağ yardımıyla "arizali_cam", "saglam_cam" ve "belirsiz" etiketlemelerini yapacağı bu resimde izolatörlerin çevrelerini taşıyarak boyamasıdır (etiketlemesi). İzolatörlerden taşan etiketler, bu üç izolatör etiketinin yüzdelik "iou" değerlerinin düşmesine neden olur. Halbuki bilgisayarımızın, eğitilmiş ağ yardımıyla etiketleme yaparken beş çeşit izolatör sınıfımızı çok daha yüksek doğrulukta bulduğu ileride bahsedeceğimiz "metrics.ClassMetrics" komutuyla oluşan "doğruluk (accuracy)" sütununda beş sınıf için gösterilmiştir.

2.7. Etiketlediğimiz Kırk Resim ile Eğitilmiş Ağımızın İzolatör Sınıflarını Doğru Etiketlemedeki Başarısının On Altı Resimlik "pxdsTest" Resim Kümesi için Ölçümü

Bu başlıktaki ölçümler, 40 resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş 16 adet resim içeren "imdsTest" resim kümesi ve ağımız eğitilmeden önce "image labeler" uygulamasıyla etiketlenerek "Label_0001.png, Label_0002.png,... Label_0040.png" formatına dönüştürülen etiketli 40 resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş 16 adet resim içeren "pxdsTest" resim kümesi baz alınarak yapılmıştır.

"imdsTest" ve "pxdsTest" resim kümelerinin 40 resmimizin kaçınıcı resimlerine denk düştüğü şekil 2.25 ve şekil 2.26'da gösterilmiştir. Her iki şekil incelendiğinde "imdsTest" ve "pxdsTest" resim kümelerinin karşılıklı olarak aynı resmi temsil ettiği görülmektedir. Mesela, bir örnekle açıklarsak; "imdsTest" resim kümesinin 3'üncü sırasındaki "DSC_8380.JPG" etiketlenmemiş resmimiz ile "pxdsTest" resim kümesinin 3'üncü sırasındaki etiketlediğimiz "Label_0015.png" resmimizin her ikisi de 40 resimlik resim klasörümüzün 15'inci resmini temsil eder.

imdsTest.Files	
26'ncı resim	1 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8507.JPG
19'uncu resim	2 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8451_porselen.JPG
15'inci resim	3 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8380.JPG
1'inci resim	4 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8243_porselen.JPG
36'ncı resim	5 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8578.JPG
23'üncü resim	6 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8494.JPG
2'nci resim	7 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8261_porselen.JPG
4'üncü resim	8 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8264.JPG
18'inci resim	9 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8413.JPG
24'üncü resim	10 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8498.JPG
39'uncu resim	11 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8614_porselen.JPG
13'üncü resim	12 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8370.JPG
9'uncu resim	13 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8311.JPG
20'nci resim	14 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8488.JPG
10'uncu resim	15 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8320.JPG
12'nci resim	16 C:\Users\Hp\AppData\Local\Temp\ \imagesResized\DSC_8360.JPG

Şekil 2.25. Kırk resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş on altı adet resim içeren “imdsTest” resim kümesinin çalışma uzayı çıktısı

pxdsTest.Files	
26'ncı resim	1 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0026.png
19'uncu resim	2 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0019.png
15'inci resim	3 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0015.png
1'inci resim	4 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0001.png
36'ncı resim	5 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0036.png
23'üncü resim	6 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0023.png
2'nci resim	7 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0002.png
4'üncü resim	8 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0004.png
18'inci resim	9 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0018.png
24'üncü resim	10 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0024.png
39'uncu resim	11 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0039.png
13'üncü resim	12 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0013.png
9'uncu resim	13 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0009.png
20'nci resim	14 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0020.png
10'uncu resim	15 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0010.png
12'nci resim	16 C:\Users\Hp\AppData\Local\Temp\ \labelsResized\Label_0012.png

Şekil 2.26. Kırk resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş on altı adet resim içeren “pxdsTest” resim kümesinin çalışma uzayı çıktısı

% On yedinci alt kod blođu (Etiketlediđimiz kırk resim ile eđitilmiş ađımız yardımıyla bilgisayarın 16 adet resim ieren "pxdsTest" resim kumesi iin izolatör etiket sınıflarını dođru bulmadaki başarı oranının kapsamlı olarak ölçülmesi)

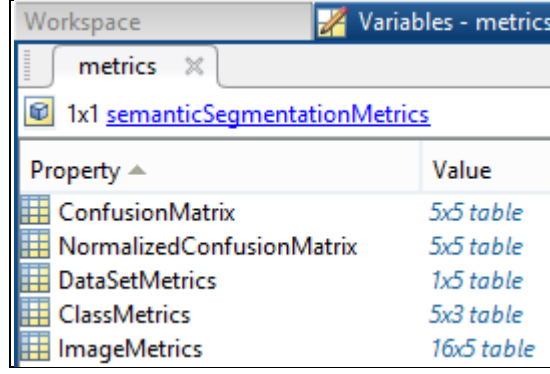
```
pxdsResults = semanticseg(imdsTest,net,'WriteLocation',tempdir,'Verbose',false);
metrics = evaluateSemanticSegmentation(pxdsResults,pxdsTest,'Verbose',false);
tbl2 = countEachLabel(pxdsTest)
metrics.ConfusionMatrix
metrics.NormalizedConfusionMatrix
metrics.DataSetMetrics
metrics.ClassMetrics
metrics.ImageMetrics
```

On yedinci alt kod blođunda; "semanticseg" fonksiyonu, daha önceden oluşturulmuş "imdsTest" veri kumesindeki 16 adet resmi ("imdsTest" veri setinin oluşturulması, sekizinci alt kod blođunda anlatılmıştı), eđitilmiş ađımız olan "net" yardımıyla piksel piksel kodlayarak (beş çeşit sınıf etiketimiz ile "Index:1, Index:2, Index:3, Index:4, Index:5" deđerleriyle kodlar) oluşturduđu 16 adet kodlanmış resmi "pxdsResults" deđişkenine atar. Ardından "semanticseg" fonksiyonu, bu 16 adet 360x480 boyutlarında ve ".png" uzantılı olarak kodlanmış resmi "tempdir" komutu yardımıyla pixelLabel_01, pixelLabel_02,... pixelLabel_16 adlarıyla "Temp" klasörüne yazdırıp bu resimleri "pxdsResults" deđişkenine atar. Bu on altı resimden oluşan "pxdsResult" resim kumesinin sıralaması da "imdsTest" ve "pxdsTest" klasörlerindeki sıralamayla aynıdır. "Temp" klasörü, genel olarak bilgisayarların işletim sisteminin yüklü olduđu "C" klasörü ierisindeki "Local" adlı klasör iinde oluşur. Bizim bilgisayarımızdan örnek verecek olursak, "Temp" klasörünün konumunun "C:\Users\...\AppData\Local\Temp" olduđu görülür. Bu kodlanmış on altı resimden oluşan "pxdsResult" resim kumesinin ilk resmi (kırk resmimizin 26'ncı resmi) iin oluşturulmuş "pixelLabel_01" resminin küçük bir parçasının piksellerinin ilk 14 satır ve ilk 21 sütunu iin piksel deđerleri örnek olarak, tablo 2.6'da gösterilmiştir.

Tablo 2.6. On altı adet kodlanmış resim içeren "pxdsResult" resim kümesinin ilk resmi (kırk resmimizin 26'ncı resmi) için oluşturulmuş "pixelLabel_01.png" resmi piksellerinin ilk 14 satır ve ilk 21 sütunu için piksel değerleri

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	3	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	5	5	5	5	5	5	5	5	5	3	5	5	5	5	5	5	5
4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5
8	5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5
9	5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5
10	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
11	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
12	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
13	5	5	5	5	5	5	5	5	5	5	5	5	5	2	5	5	3
14	5	5	5	5	5	5	5	5	5	5	5	5	5	1	3	5	5

On yedinci alt kod bloğunun ikinci satırındaki "evaluateSemanticSegmentation" fonksiyonu ise "imdsTest" klasöründen alınarak "net" ağı ile kodlanmış ve "pxdsResults" değişkenine atanmış bu 16 adet kodlanmış resim ile etiketlemelerini ağ eğitilmeden önce yaptığımız "pxdsTest" veri kümesindeki 16 adet resmi karşılaştırır ("pxdsTest" veri setinin oluşturulması, sekizinci alt kod bloğunda anlatılmıştı). Bu karşılaştırma sonucunda hesapladığı doğruluk başarımlarını "metrics" değişkenine atar. "metrics" değişkeninin şekil 2.27'deki çalışma uzayı görünümünden de anlaşıldığı üzere içerisinde "karışıklık matrisi (ConfusionMatrix)", "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)", "veri kümesi ölçümleri (DataSetMetrics)", "sınıf ölçümleri (ClassMetrics) ve "resim ölçümleri (ImageMetrics)" olmak üzere beş farklı doğruluk başarımlarını ölçümü yaklaşımı mevcuttur.



Property	Value
ConfusionMatrix	5x5 table
NormalizedConfusionMatrix	5x5 table
DataSetMetrics	1x5 table
ClassMetrics	5x3 table
ImageMetrics	16x5 table

Şekil 2.27. "evaluateSemanticSegmentation" fonksiyonu ile oluşturulan "metrics" değişkeninin çalışma uzayı görünümü

Tablo 2.7'deki "tbl2" tablosu, 360x480 boyutlarına düşürülmüş "pxdsTest" veri seti içerisindeki 16 adet resim için, her bir etiketin kapladığı toplam piksel sayısını hesaplayarak "PixelCount" sütununda; ilgili etiket sınıfını içeren resimlerin toplam piksel sayısını hesaplayarak "ImagePixelCount" sütununda gösterir. Tablo 2.8'deki "karışıklık matrisi (ConfusionMatrix)" tablosu, etiketlediğimiz piksellerin bilgisayar tarafından hangi sınıf olarak algılandığını karşılaştırır. Örneğin, "karışıklık matrisi" tablosunun ilk satırında, "arizali_cam" olarak etiketlediğimiz piksellerden 4664 adedi "arizali_cam" olarak, 3 adedi "arizali_pors" olarak, 7348 adedi "saglam_cam" olarak, 0 adedi "saglam_pors" olarak ve 4692 adedi de "belirsiz" olarak eğitilmiş ağıımız tarafından algılanmış. Dikkat edilirse, şekil 2.35'deki tablonun ilk satırındaki "arizali_cam" etiketi için "PixelCount" değeri (16707) ile şekil 2.36'daki tablonun ilk satırındaki "arizali_cam" etiketi için verilen piksel değerlerinin toplamının ($4664+3+7348+0+4692=16707$) birbirine eşit olduğu görülür. Bu örnekte anlattığımız işlemler, tablo 2.7 ve tablo 2.8'deki tabloların diğer dört satırındaki etiketlerimiz için de geçerlidir.

Tablo 2.7. Resimlerimizin boyutları 360x480 iken on altı resimlik “pxdsTest” resim kümesi için etiketlerimizin piksel sayılarının (PixelCount) ve etiketlerimizin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi

Sınıf (Etiket) Adı	Etiket Piksel Sayıları (PixelCount)	Etiketlerin Bulunduğu Resimlerin Toplam Piksel Sayıları (ImagePixelCount)
'arizali_cam'	16707	2073600
'arizali_pors'	4310	691200
'saglam_cam'	37017	2419200
'saglam_pors'	17502	1555200
'belirsiz'	15791	1728000

Tablo 2.8. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu

	'arizali_cam'	'arizali_pors'	'saglam_cam'	'saglam_pors'	'belirsiz'
'arizali_cam'	4664	3	7348	0	4692
'arizali_pors'	95	62	219	2542	1392
'saglam_cam'	8240	2	21608	0	7167
'saglam_pors'	484	91	1942	12944	2041
'belirsiz'	120	57	5289	956	9369

Tablo 2.9'daki "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)" tablosu ise etiketlediğimiz piksellerin bilgisayar tarafından hangi sınıf olarak algılandığının yüzdesini her bir sınıf için ayrı ayrı verir. Örneğin, "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)" tablosunun ilk satırında, “arizali_cam” olarak etiketlediğimiz 16707 adet pikselin %27,916’sı “arizali_cam” olarak, %0,017957’si “arizali_pors” olarak, %43,982’si “saglam_cam” olarak, %0’ı “saglam_pors” olarak ve %28,084’ü de ”belirsiz” olarak eğitilmiş ağımız tarafından algılanmış. Bu tablonun köşegen matris yüzde değerleri incelendiğinde “pxdsTest” resim kümesi (40 resmimizin 16 tanesi) için eğitilmiş ağımızın, “arizali_cam” etiketini bulma başarısı %27,916, “arizali_pors” etiketini bulma başarısı %1,4385, “saglam_cam” etiketini bulma başarısı %58,373, “saglam_pors” etiketini bulma başarısı %73,957 ve “belirsiz” etiketini bulma başarısı da %59,331’dir. Bu sayılan yüzdeler başarı oranları, aynı zamanda Tablo 2.11'deki “sınıf ölçümleri (ClassMetrics)”

tablosunun “doğruluk (Accuracy)” sütununda da görülebilir. Tablo 2.8’deki “karışıklık matrisi (ConfusionMatrix)” tablosundan yararlanılarak tablo 2.9’daki “0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)” tablosunun köşegen matrisindeki yüzdesel başarımların bulunması aşağıda gösterilmiştir.

$$\text{“arizali_cam”}: [(4664) \div (4664+3+7348+0+4692)] \times 100 = \%27,916$$

$$\text{“arizali_pors”}: [(62) \div (95+62+219+2542+1392)] \times 100 = \%1,4385$$

$$\text{“sağlam_cam”}: [(21608) \div (8240+2+21608+0+7167)] \times 100 = \%58,373$$

$$\text{“sağlam_pors”}: [(12944) \div (484+91+1942+12944+2041)] \times 100 = \%73,957$$

$$\text{“belirsiz”}: [(4664) \div (120+57+5289+956+9369)] \times 100 = \%59,331$$

Tablo 2.9. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan “0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)” tablosu

	'arizali_cam'	'arizali_pors'	'sağlam_cam'	'sağlam_pors'	'belirsiz'
arizali_cam	0,2792	0,0002	0,4398	0,0000	0,2808
arizali_pors	0,0220	0,0144	0,0508	0,5898	0,3230
sağlam_cam	0,2226	0,0001	0,5837	0,0000	0,1936
sağlam_pors	0,0277	0,0052	0,1110	0,7396	0,1166
belirsiz	0,0076	0,0036	0,3349	0,0605	0,5933

Tablo 2.10’deki “veri kümesi ölçümleri (DataSetMetrics)” tablosunun “MeanIoU” sütunundaki değer, tablo 2.11’deki “sınıf ölçümleri (ClassMetrics)” tablosunda gösterilmiş olan beş etiketimiz için hesaplanmış “IoU” yüzdesel başarımların (%1,5452; %0,049877; %5,4179; %2,1066; %0,68484) aritmetik ortalamasıdır; “ortalama doğruluk (MeanAccuracy)” sütunundaki değer ise “NormalizedConfusionMatrix” tablosunun köşegen matrisinde gösterilen beş ayrı sınıfın tespit edilmesindeki başarımların veya tablo 2.11’de gösterilmiş olan beş etiketimiz için hesaplanmış “Accuracy” yüzdesel başarımların (%27,916; %1,4385; %58,373; %73,957; %59,331) aritmetik ortalamasıdır. Bu tablodaki “GlobalAccuracy” değeri ise tüm doğru bulunan piksel sayısının, etikediğimiz tüm piksellerin sayısına yüzdesel oranıdır; yani “ConfusionMatrix” tablosunun köşegen matris piksel değerlerinin (4664, 62, 21608, 12944, 9369) toplamının, “tbl2” tablosundaki “PixelCount” değerlerinin (16707, 4310, 37017, 17502, 15791) toplamına oranının yüzdesidir. Bu hesaplama aşağıda gösterilmiştir.

$$\text{GlobalAccuracy} = \frac{4664 + 62 + 21608 + 12944 + 9369}{16707 + 4310 + 37017 + 17502 + 15791} \times 100 = \%53,27$$

Tablo 2.10. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan "veri kümesi ölçümleri (DataSetMetrics)" tablosu

Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
0,5327	0,4420	0,0196

Tablo 2.11’deki “IoU” değerlerinin bulunması, on altıncı alt kod bloğunda anlatılmıştı; “Accuracy” değerlerinin bulunması ise tablo 2.9’daki “NormalizedConfusionMatrix” tablosunun köşegen matrisindeki yüzdesel başarımların bulunması kısmında anlatılmıştı (tablo 2.9’daki köşegen matris değerleri ile tablo 2.11’de gösterilmiş olan “Accuracy” değerlerinin aynı olduğu görülmektedir). Burada, “Accuracy” ve “IoU” değerlerinin her ikisinin de eğitilmiş ağın etiketlediğimiz pikselleri doğru algılamadaki başarımların ölçütleri olduğunu söylemekle birlikte aralarındaki farkı bir örnekle açıklayalım. Önce iki küme belirleyelim, “A” kümesi bizim etiketlediğimiz pikseller olsun, “B” kümesi ise bilgisayarın etiketlediği pikseller olsun. “IoU” değeri, bu iki kümenin kesişim kümesinin eleman sayısının, birleşim kümesinin eleman sayısına yüzdesel oranıdır; “Accuracy” değeri ise bu iki kümenin kesişim kümesinin eleman sayısının, “A” kümesinin eleman sayısına yüzdesel oranıdır. Bu iki değerin matematiksel ifadeleri aşağıda gösterilmiştir.

$$\text{“Accuracy(A,B) = [s(A} \cap \text{B) / s(A)]x100”}$$

$$\text{“IoU(A,B) = [s(A} \cap \text{B) / s(A} \cup \text{B)]x100”}$$

Tablo 2.11. On altı resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu

	Doğruluk (Accuracy)	iou (birleşimlerinin kesişimi)
'arizali_cam'	0,2792	0,0155
'arizali_pors'	0,0144	0,0005
'saglam_cam'	0,5837	0,0542
'saglam_pors'	0,7396	0,0211
'belirsiz'	0,5933	0,0068

Tablo 2.12'deki "resim ölçümleri (ImageMetrics)" tablosunda 16 resimlik “imdsTest” veri kümesi resimlerin her biri için ayrı ayrı yüzdesel başarımları hesaplanmıştır. Bunlardan “GlobalAccuracy”, “MeanAccuracy” ve “MeanIoU” değerlerinin hesaplanmasını bir örnekle açıklayalım. Beş ayrı etiketimiz için bizim etiketlediğimiz pikselleri içeren kümeler A, B, C, D, E kümeleri olsun; bunlara karşılık gelen bilgisayarın (eğitilmiş ağıımızın) etiketlediği pikselleri içeren kümeler ise A1, B1, C1, D1, E1 kümeleri olsun. “GlobalAccuracy” değeri, beş sınıf için bizim etiketlediğimiz pikseller ile bilgisayarın etiketlediği piksellerin örtüştüğü (kesiştği) piksel sayılarının toplamının, bizim etiketlediğimiz piksel sayılarının toplamına yüzdesel oranıdır. “MeanAccuracy” değeri, beş sınıf için bizim etiketlediğimiz pikseller ile bilgisayarın etiketlediği piksellerin örtüştüğü (kesiştği) piksel sayılarının, her bir sınıf etiketi için ayrı ayrı bizim etiketlediğimiz piksellerin sayısına bölümlerinin aritmetik ortalamasıdır. “MeanIoU” değeri ise beş sınıf için bizim etiketlediğimiz pikseller ile bilgisayarın etiketlediği piksellerin örtüştüğü (kesiştği) piksel sayılarının, her bir sınıf etiketi için ayrı ayrı bizim etiketlediğimiz pikseller ile bilgisayarın etiketlediği piksellerin birleşim kümesi eleman sayısına bölümlerinin aritmetik ortalamasıdır. Bu üç değerın matematiksel gösterimi aşağıda gösterilmiştir.

$$\text{GlobalAccuracy} = \frac{s(A \cap A1) + s(B \cap B1) + s(C \cap C1) + s(D \cap D1) + s(E \cap E1)}{s(A) + s(B) + s(C) + s(D) + s(E)} \times 100$$

$$\text{MeanAccuracy} = \left(\frac{s(A \cap A1)}{s(A)} + \frac{s(B \cap B1)}{s(B)} + \frac{s(C \cap C1)}{s(C)} + \frac{s(D \cap D1)}{s(D)} + \frac{s(E \cap E1)}{s(E)} \right) / 5$$

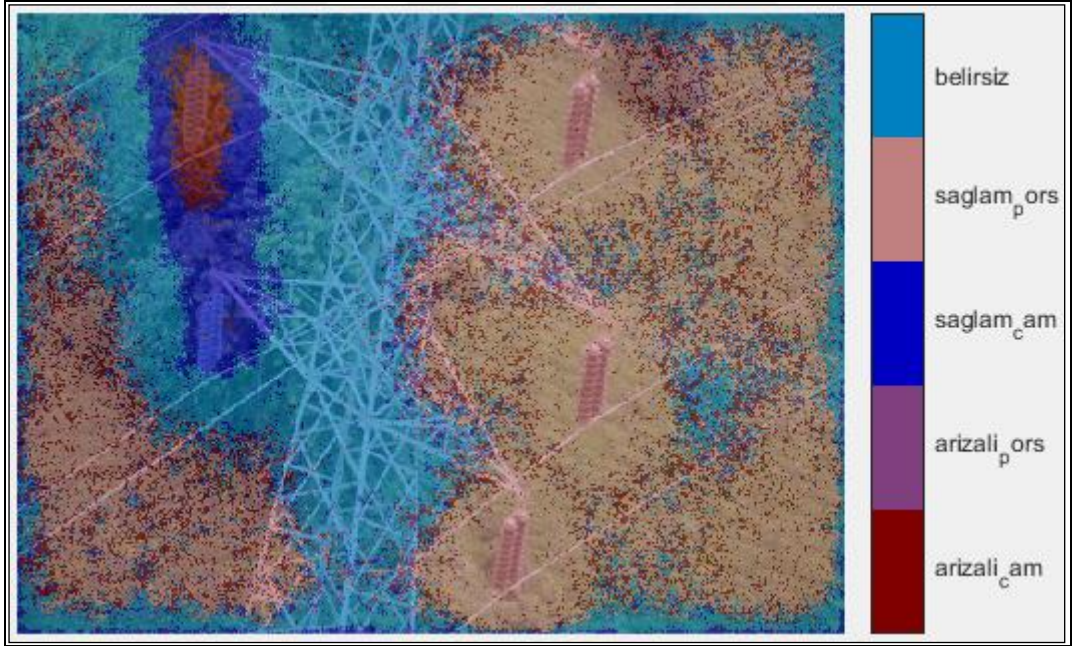
$$\text{MeanIoU} = \left(\frac{s(A \cap A1)}{s(A \cup A1)} + \frac{s(B \cap B1)}{s(B \cup B1)} + \frac{s(C \cap C1)}{s(C \cup C1)} + \frac{s(D \cap D1)}{s(D \cup D1)} + \frac{s(E \cap E1)}{s(E \cup E1)} \right) / 5$$

Tablo 2.12. On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu

	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
1	0,5056	0,5647	0,0157
2	0,5234	0,4301	0,0105
3	0,5014	0,3478	0,0199
4	0,6377	0,5195	0,0195
5	0,8981	0,8855	0,0170
6	0,3988	0,2810	0,0122
7	0,5964	0,4915	0,0136
8	0,7482	0,7644	0,0274
9	0,8029	0,7580	0,0190
10	0,3502	0,3286	0,0063
11	0,1099	0,0944	0,0040
12	0,9712	0,9477	0,0244
13	0,4086	0,3411	0,0187
14	0,5492	0,4750	0,0219
15	0,7578	0,7510	0,0204
16	0,5842	0,3298	0,0252



(a)

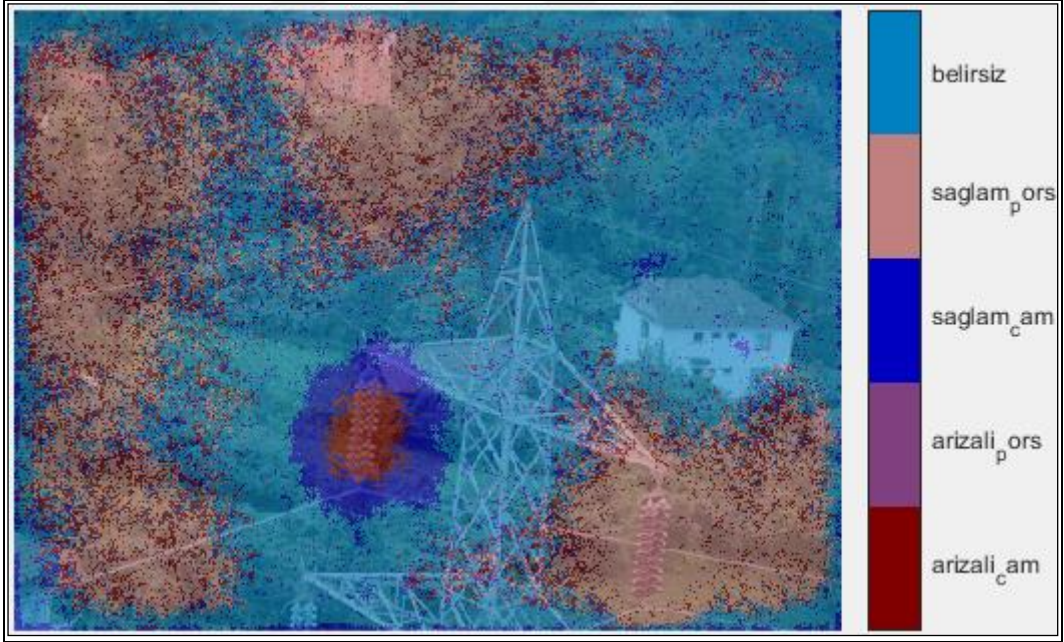


(b)

Şekil 2.28. On altı resimlik “imdsTest” resim kümesinin 12’nci resmi (40 resmimizin 13’üncü resmi) için , (a) Eğitim öncesi, etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

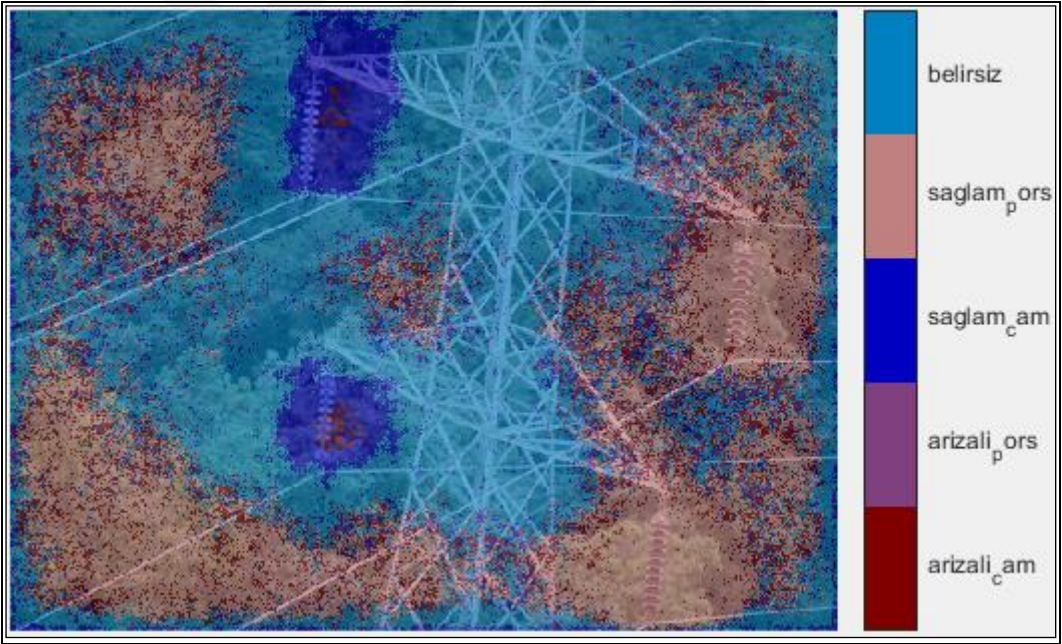


(b)

Şekil 2.29. On altı resimlik “imdsTest” resim kümesinin 5’inci resmi (40 resmimizin 36’ncı resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

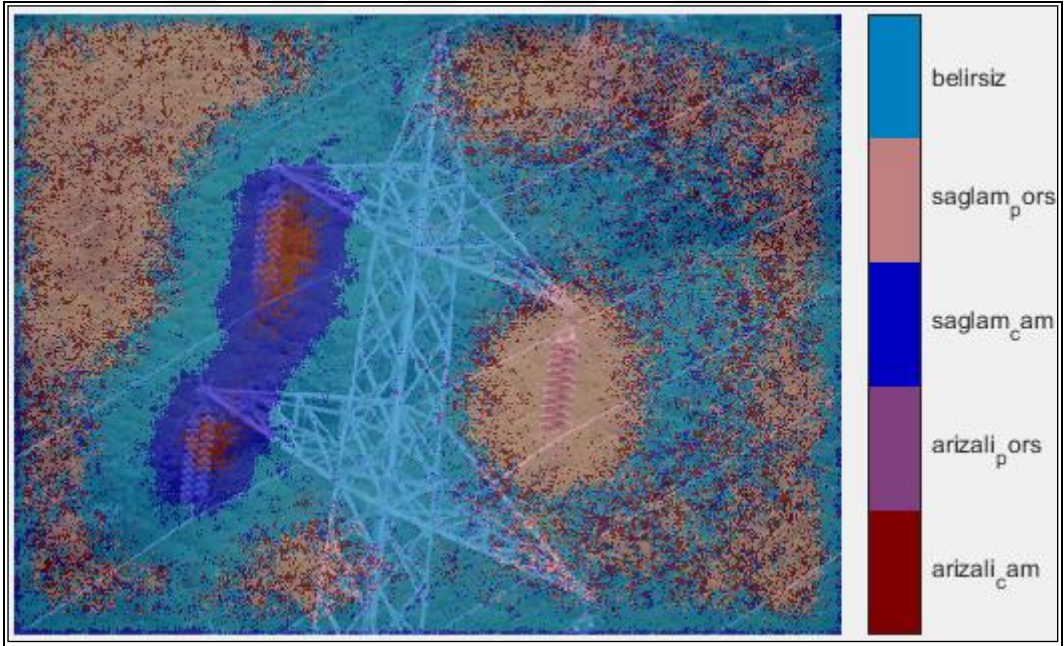


(b)

Şekil 2.30. On altı resimlik “imdsTest” resim kümesinin 9’uncu resmi (40 resmimizin 18’inci resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

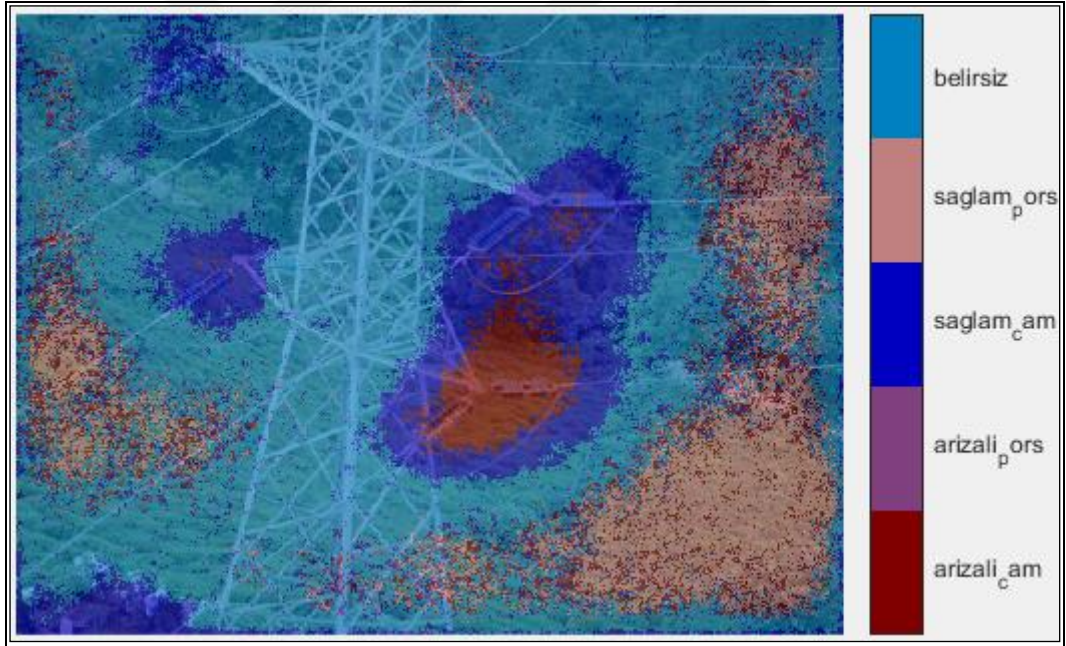


(b)

Şekil 2.31. On altı resimlik “imdsTest” resim kümesinin 15’inci resmi (40 resmimizin 10’uncu resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)



(b)

Şekil 2.32. On altı resimlik “imdsTest” resim kümesinin 8’inci resmi (40 resmimizin 4’üncü resmi) için , (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Tablo 2.13. Kırk resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.28.b, şekil 2.29.b, şekil 2.30.b, şekil 2.31.b ve şekil 2.32.b'de gösterilenlerinin yüzdesel doğruluk başarımlarını karşılaştırması

	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
Şekil 2.28.b	%97,12	%94,77	%2,44
Şekil 2.29.b	%89,81	%88,55	%1,70
Şekil 2.30.b	%80,29	%75,80	%1,90
Şekil 2.31.b	%75,78	%75,10	%2,04
Şekil 2.32.b	%74,82	%76,44	%2,74

% On sekizinci alt kod bloğu (Tablo 2.12'deki on altı resimlik "imdsTest" ve "pxdsTest" resim kümeleri için hesaplanmış "ImageMetrics" tablosundaki "MeanIoU" değerlerinin bulunması)

```
I3 = readimage(imdsTest,1);
```

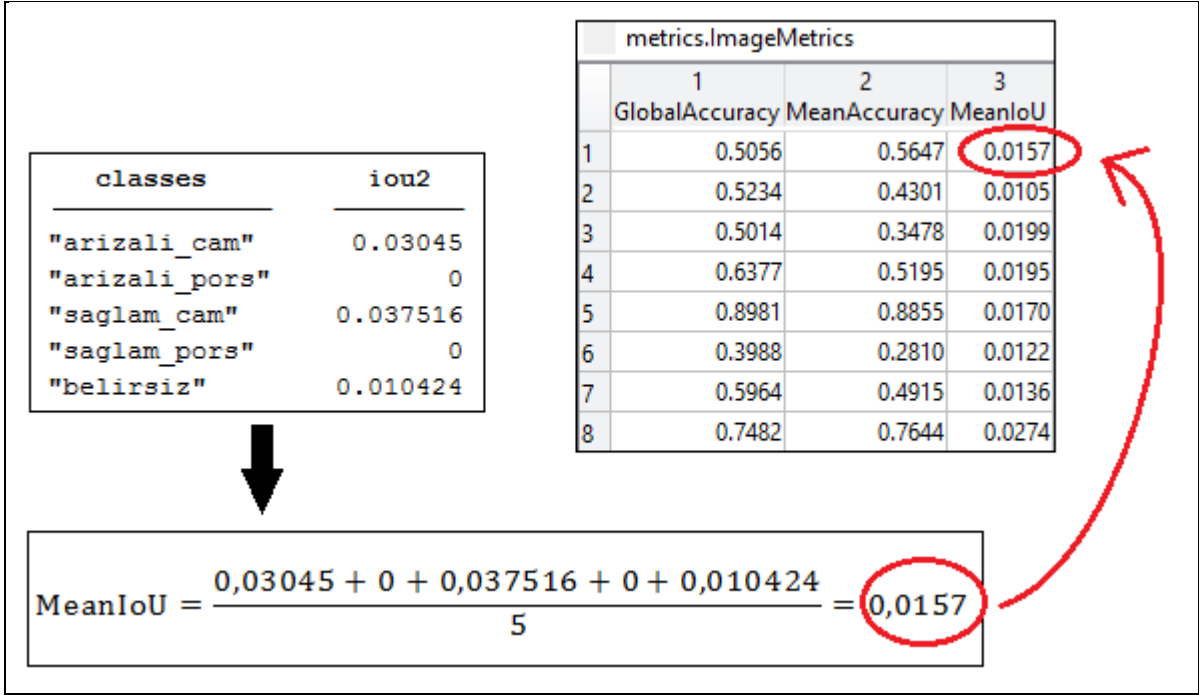
```
C3 = semanticseg(I3, net);
```

```
expectedResult2 = readimage(pxdsTest,1);
```

```
iou2 = jaccard(C3,expectedResult2);
```

```
table(classes,iou2)
```

On sekizinci alt kod bloğunu kullanarak, tablo 2.12'deki on altı resimlik "imdsTest" ve "pxdsTest" resim kümeleri için hesaplanmış "ImageMetrics" tablosunda her resim için verilmiş "MeanIoU" değerlerinin nasıl elde edildiğini gösterelim. Örneğin, on altı resimlik "pxdsTest" resim kümesinin ilk resmini (40 resmimizin 26'ncı resmini) baz alalım. On sekizinci alt kod bloğumuz çalıştırıldığında, bu resmin beş sınıfı için "iou" değerlerini elde ederiz. Bu değerlerin aritmetik ortalaması alındığında "MeanIoU" değerini elde etmiş oluruz. Yapılan bu işlemler şekil 2.33'te gösterilmiştir.



Şekil 2.33. On altı resimlik "imdsTest" resim kümesinin ilk resmi (40 resmimizin 26'ncı resmi) için "iou" değerlerinin elde edilmesi ve "MeanIoU" değerinin ispatı

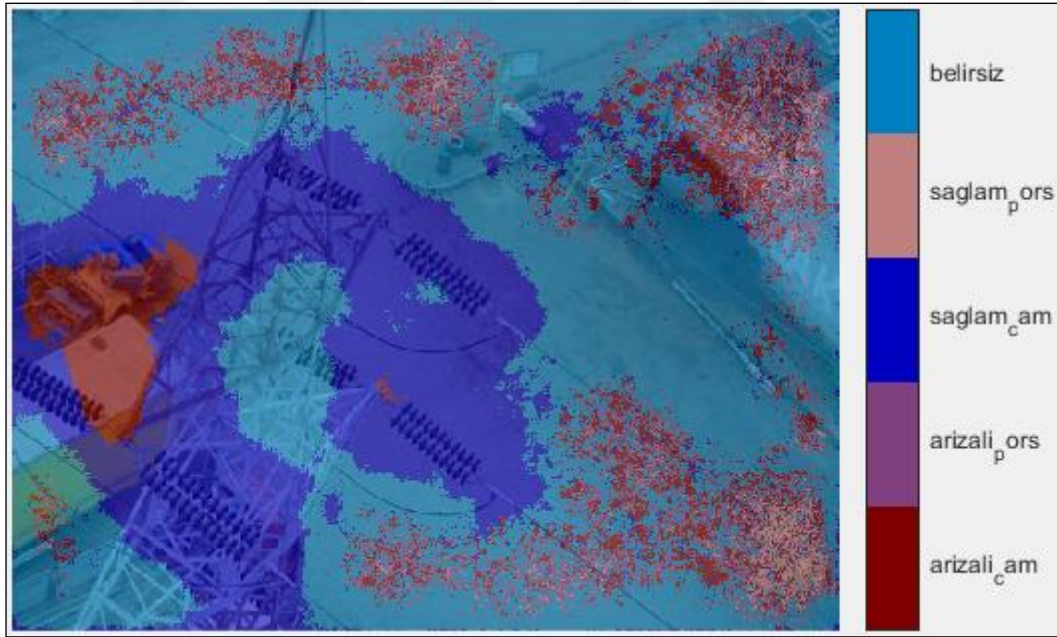
Tablo 2.12'deki on altışar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için "ImageMetrics" tablosunda her resim için hesaplanan yüzdesel doğruluk başarımları değerlendirildiğinde "MeanIoU" değerlerinin, "GlobalAccuracy" ve "MeanAccuracy" değerlerine göre çok daha düşük olduğu görülecektir. Bunun sebeplerine değinelim. Birinci sebebi, "MeanIoU" veya "iou" değerleri hesaplanırken eğitilmiş ağımızın, tespit etmesini istediğimiz izolatörlerin çevrelerini taşıyarak etiketlemesidir. İkinci sebebi, eğitilmiş ağımızın, bizim etiketlemediğimiz alanları da etiketlemesidir (eğitilmiş ağımız, resimde hiçbir boşluk bırakmaksızın istisnasız tüm pikselleri etiketler). Üçüncü sebebi açıklayalım: "Herhangi bir resimde etiketlediğimiz izolatör sınıfı olmadığı durumda kesişim boş küme olacağından "iou" değeri sıfır çıkacaktır; ardından "MeanIoU" değeri bulunurken her sınıf için çıkan bu sıfır "iou" değerleri, aritmetik ortalamaya dahil edileceğinden "MeanIoU" değerini fazlasıyla aşağı çekecektir". Biz burada, etiketlediğimiz izolatörler sınıflarını, eğitilmiş ağımızın doğru bulabilme başarısını ölçmek istediğimiz için "GlobalAccuracy" ve "Accuracy" değerlerini baz alırız; çünkü bizi etiketlediğimiz izolatörlerin dışındaki pikseller değil, etiketlediğimiz izolatörler üzerindeki pikseller ilgilendiriyor. Şayet, "MeanIoU" ve "IoU" değerlerini yükseltmek isteseydik, ağımızı eğitmeden önce etiketlediğimiz izolatörlerin dışındaki tüm pikselleri de etiketlememiz gerekirdi (mesela; ağaç, gözyüzü, bina, yol, vb.). Bu şekilde eğiteceğimiz ağımız, izolatörlerin çevresini taşımadan

etiketleme yaparak izolatörlerin dışındaki alanları bu yeni sınıflardan uygun olanları ile etiketleyecek; bunun sonucunda da "Mean IoU" ve "iou" değerlerini çok daha yüksek doğruluk oranlarıyla hesaplayabilecekti. Bu çalışmamız için bu gereksiz olurdu, çünkü az önce de bahsettiğimiz gibi bizi izolatörlerin dışındaki pikseller değil izolatörlerin ne derece doğru algılandığı ilgilendiriyor, bunu veren sonuçlar da "GlobalAccuracy", "MeanAccuracy" ve "Accuracy" değerleridir.

Tablo 2.12'deki on altışar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için "ImageMetrics" tablosunun 11'inci satırı (40 resmimizin 39'uncu resmi) için "GlobalAccuracy" ve "MeanAccuracy" değerlerinin diğer satırlara göre çok daha düşük olma sebebi, eğitilen ağın 39'uncu resimdeki izolatörleri doğru algıyalamamış olmasıdır, çünkü 39'uncu resimdeki porselen izolatörlerin görüntüleri diğer resimlerdeki porselen izolatörlerden farklı olarak sadece bu resimde yatay pozisyonadadır (diğer resimlerdeki porselen izolatörlerin hepsi dikey pozisyonadadır, örneğin şekil 2.28.a, şekil 2.29.a, şekil 2.30.a ve şekil 2.31.a'da gösterildiği gibi). Ağımız eğitilmeden önce, kırk resim için etiketlediğimiz izolatör sınıfları arasında yatay pozisyona sahip olanlar sadece sağlam cam izolatör sınıfı ile arızalı cam izolatör sınıfı olduğu ve bu iki sınıf arasında en fazla etiketlenmiş piksele sahip olan sınıfın ise sağlam cam izolatör sınıfı olduğu göz önüne alındığında ağımız, eğitimden sonra resimleri etiketlerken baskın sınıfların lehine ön yargılı olarak etiketleme yapacağı için bu yatay pozisyonadaki porselen izolatörlerin büyük çoğunluğunu sağlam cam izolatör sınıfı ile etiketler (şekil 2.34). Ağımızı eğitmek için kullanacağımız fotoğraflar arasında 39'uncu resmin benzerlerinin sayısı daha fazla olsaydı, eğitilmiş ağımız bu resim için çok daha yüksek "GlobalAccuracy" ve "MeanAccuracy" değerleri bulurdu fakat hat boyunca çekilen fotoğraflarımızın çeşit yelpazesi sınırlı.



(a)



(b)

Şekil 2.34. On altı resimlik “imdsTest” resim kümesinin 11’inci resmi (40 resmimizin 39’uncu resmi) için, (a) Eğitim öncesi, etiklediğimiz resim, (b) Kırk resimle eğitilmiş ağ kullanılarak bilgisayarın, resimleri etiketlerken baskın sınıfların lehine yatay pozisyondaki porselen izolatörlerin büyük çoğunluğunu sağlam cam izolatör sınıfıyla etiketlemesi

Tablo 2.11’deki on altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resim ile eğitilmiş ağ kullanılarak oluşturulan "ClassMetrics" tablosunda beş sınıf için "Accuracy" değerleri verilmiştir. Bu değerler incelendiğinde, eğitilmiş ağımızın, arızalı

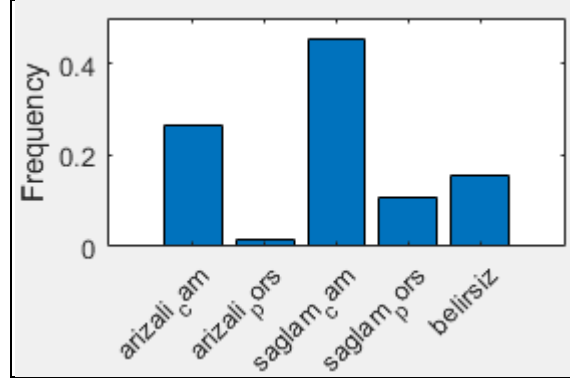
porcelenleri algılamadaki başarısının diğere dört sınıfa göre çok daha düşük olduđu görüldü. Bunun sebebi, bir önceki paragrafta bahsettiğimiz gibi ağır baskın sınıfların lehine ön yargılı olarak etiketleme yapmasıdır; çünkü etiketlediğimiz arızalı porcelenlerin piksel sayısı, diğere dört sınıftan çok daha düşüktür (on altı resimlik "pxdsTest" klasöründe beş sınıf için etiketlediğimiz sınıfların piksel sayısı dağılımları Tablo 2.7'deki "tbl2" tablosunun "PixelCount" sütununda görülmektedir). Eğitilmiş ağıımızın arızalı porcelenleri algılamadaki başarısını artırmak için daha fazla arızalı porcelenin ağıya tanıtılması gerekirdi. Bir sonraki başlıkta, ağıımızı eğiteceğimiz etiketli resim sayımızı artırdığımızda (40'tan 100'e çıkararak) etiketlediğimiz arızalı porcelen piksel sayısı da artacak, bunun sonucunda da eğitilmiş ağıımızın arızalı porcelen sınıflarını algılamadaki başarısının ("Accuracy" değerlerinin) yaklaşık 3 katına yükseldiği görülecektir (%1,44'ten %4,48'ye).

2.8. Etiketlediğimiz Yüz Resim ile Ağıımızın Eğitimi

Bu başlıkta, etiketlediğimiz resim sayısının 40'tan 100'e çıkarılarak ağıımızın yeniden eğitilmesi anlatılmıştır. Aynı matlab kod bloğumuzu kullanacağımız için kodlar tekrar uzun uzadıya yazılmamıştır. Resim klasörümüzün (imds) ve etiketli resim klasörümüzün (pxds) içerisine yeni resimler ekleneceği için matlab kod bloğumuzda değişen sadece, "images_40resim" yerine "images_100resim"; "labels_40resim" yerine de "labels_100resim" yazılması olacaktır. 100 resmimiz için "imdsTrain" ve "pxdsTrain" veri kümeleri, %60'a karşılık gelen 60'ar adet resim içerirken "imdsTest" ve "pxdsTest" veri kümeleri ise %40'a karşılık gelen 40'ar adet resim içerir. Etiketlediğimiz 100 resim ile ağır eğitilmesi, "92194,52" saniye yani yaklaşık 26 saat sürmüştür (40 resim ile yaklaşık 11 saat sürmüştü).

Tablo 2.14. Resimlerimizin boyutları 4912x7360 iken yüz resmimiz için etiket piksel sayılarının (PixelCount) ve etiketlerin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi

Sınıf (Etiket) Adı	Etiket Piksel Sayıları (PixelCount)	Etiketlerin Bulunduğu Resimlerin Toplam Piksel Sayıları (ImagePixelCount)
'arizali_cam'	34440301	3253708800
'arizali_pors'	1908877	469980160
'saglam_cam'	58647155	3362165760
'saglam_pors'	14145170	1446092800
'belirsiz'	20430865	2458357760



Şekil 2.35. Etiketlediğimiz yüz resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu piksel sayılarının yüzdelik dağılımlarının grafiksel gösterimi

Tablo 2.15. Etiketlediğimiz kırk resimlik “pxds” resim kümemiz için etiketlerin sahip olduğu sınıf etiketleri piksellerinin sıklık yüzdesi

Sınıf (Etiket) Adı	Sınıf Etiketleri Pksellerinin Sıklık Yüzdesi (frequency)
'arizali_cam'	0,2658
'arizali_pors'	0,0147
'saglam_cam'	0,4526
'saglam_pors'	0,1092
'belirsiz'	0,1577

Tablo 2.16. Etiketlediğimiz yüz resimlik “pxds” resim kümemiz için sınıf ağırlıkları (classWeights) ve frekans (sınıf etiketleri piksellerinin sıklık yüzdesi) karşılaştırması

Sınıf (Etiket) Adı	Sınıf Ağırlıkları (classWeights)	Sınıf Etiketleri Pksellerinin Sıklık Yüzdesi (frequency)
'arizali_cam'	0,9241	0,2658
'arizali_pors'	2,4083	0,0147
'saglam_cam'	0,5608	0,4526
'saglam_pors'	1,0000	0,1092
'belirsiz'	1,1770	0,1577

2.9. Etiketlediğimiz Yüz Resimle Eğitilmiş Ağımızın İzolatör Sınıflarını Doğru Etiketlemedeki Başarısının Kırk Resimlik Test Kümesi İçin Ölçümü

Bu başlıktaki ölçümler, 100 resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş 40 adet resim içeren “imdsTest” resim kümesi ve ağımız eğitilmeden önce “image labeler” uygulamasıyla etiketlenerek “Label_0001.png, Label_0002.png,... Label_0100.png” formatına dönüştürülen etiketli 100 resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş 40 adet resim içeren “pxdsTest” resim kümesi baz alınarak yapılmıştır.

“imdsTest” ve “pxdsTest” resim kümelerinin, 100 resmimizin kaçınıcı resimlerine denk düştüğü şekil 2.36 ve şekil 2.37’de gösterilmiştir. Her iki şekil incelendiğinde “imdsTest” ve “pxdsTest” resim kümelerinin karşılıklı olarak aynı resmi temsil ettiği görülür. Bir örnekle açıklarsak; “imdsTest” resim kümesinin 5’inci sırasındaki “DSC_8760.JPG” etiketlenmemiş resmimiz ile “pxdsTest” resim kümesinin 5’inci sırasındaki etiketlediğimiz “Label_0064.png” resmimizin her ikisi de 100 resimlik resim klasörümüzün 64’üncü resmini temsil eder.

Şekil 2.36 ve şekil 2.37’deki yüz resim ile eğittiğimiz ağ için bilgisayarın oluşturduğu kırkar resimlik “imdsTest” ve “pxdsTest” klasörlerindeki resimlerin koyu yazılı ve altları çizili olanlarının, şekil 2.25 ve şekil 2.26’daki kırk resim ile eğittiğimiz ağ için bilgisayarın oluşturduğu on altışar resimlik “imdsTest” ve “pxdsTest” klasörlerindeki resimler ile aynı numaraya sahip oldukları görülmüştür. Bilgisayarın test için rastgele seçtiği %40’lık resim kümelerinde, 40 resim için oluşturulmuş “imdsTest” resim kümesindeki 16 resmin hepsinin, 100 resim için oluşturulmuş “imdsTest” resim kümesindeki 40 resmin içerisinde olduğu görülmüştür. Bu da bize, bilgisayarın rastgele seçeceği resim numaralarının önceden belli olduğunu gösterir. Bunun, resimlerin artırılarak ağ eğitimi yapılması çalışmalarında önceki ve sonraki çalışmaları birbirleriyle kıyasalamak açısından faydalı olacağı da yadsınamaz bir gerçektir. Bu durum, şekil 3.1’de, aynı resimler için, resim sayısı artırılarak (40’tan 100’e çıkarılarak) yapılan eğitim sonucunda bulunan doğruluk başarımlarını değerlerini kıyaslamada kullanılmıştır.

		imdsTest.Files	
21'inci resim	1	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8489.JPG
25'inci resim	2	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8502.JPG
52'nci resim	3	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8665.JPG
37'nci resim	4	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8580.JPG
64'üncü resim	5	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8760.JPG
31'inci resim	6	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8528.JPG
49'uncu resim	7	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8639.JPG
27'nci resim	8	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8510.JPG
61'inci resim	9	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8734.JPG
88'inci resim	10	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8923.JPG
50'nci resim	11	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8640.JPG
87'nci resim	12	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8907.JPG
<u>26'nci resim</u>	13	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8507.JPG
43'üncü resim	14	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8622.JPG
94'üncü resim	15	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8973.JPG
<u>19'uncu resim</u>	16	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8451.JPG
44'üncü resim	17	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8623.JPG
<u>15'inci resim</u>	18	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8380.JPG
73'üncü resim	19	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8792.JPG
<u>1'inci resim</u>	20	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8243.JPG
<u>36'nci resim</u>	21	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8578.JPG
82'nci resim	22	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8882.JPG
71'inci resim	23	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8789.JPG
<u>23'üncü resim</u>	24	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8494.JPG
65'inci resim	25	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8764.JPG
<u>2'nci resim</u>	26	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8261.JPG
<u>4'üncü resim</u>	27	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8264.JPG
<u>18'inci resim</u>	28	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8413.JPG
85'inci resim	29	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8900.JPG
75'inci resim	30	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8815.JPG
<u>24'üncü resim</u>	31	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8498.JPG
95'inci resim	32	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8978.JPG
<u>39'uncu resim</u>	33	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8614.JPG
<u>13'üncü resim</u>	34	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8370.JPG
<u>9'uncu resim</u>	35	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8311.JPG
66'nci resim	36	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8766.JPG
<u>20'nci resim</u>	37	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8488.JPG
57'nci resim	38	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8692.JPG
<u>10'uncu resim</u>	39	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8320.JPG
<u>12'nci resim</u>	40	C:\Users\Hp\AppData\Local\Temp\	\images_100resimResized\DSC_8360.JPG

Şekil 2.36. Yüz resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş kırk adet resim içeren “imdsTest” resim kümesinin çalışma uzayı çıktısı

		pxdsTest.Files	
21'inci resim	1	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0021.png
25'inci resim	2	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0025.png
52'nci resim	3	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0052.png
37'nci resim	4	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0037.png
64'üncü resim	5	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0064.png
31'inci resim	6	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0031.png
49'uncu resim	7	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0049.png
27'nci resim	8	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0027.png
61'inci resim	9	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0061.png
88'inci resim	10	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0088.png
50'nci resim	11	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0050.png
87'nci resim	12	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0087.png
26'nci resim	13	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0026.png
43'üncü resim	14	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0043.png
94'üncü resim	15	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0094.png
19'uncu resim	16	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0019.png
44'üncü resim	17	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0044.png
15'inci resim	18	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0015.png
73'üncü resim	19	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0073.png
1'inci resim	20	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0001.png
36'nci resim	21	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0036.png
82'nci resim	22	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0082.png
71'inci resim	23	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0071.png
23'üncü resim	24	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0023.png
65'inci resim	25	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0065.png
2'nci resim	26	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0002.png
4'üncü resim	27	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0004.png
18'inci resim	28	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0018.png
85'inci resim	29	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0085.png
75'inci resim	30	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0075.png
24'üncü resim	31	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0024.png
95'inci resim	32	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0095.png
39'uncu resim	33	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0039.png
13'üncü resim	34	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0013.png
9'uncu resim	35	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0009.png
66'nci resim	36	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0066.png
20'nci resim	37	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0020.png
57'nci resim	38	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0057.png
10'uncu resim	39	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0010.png
12'nci resim	40	C:\Users\Hp\AppData\Local\Temp\	\labels_100resimResized\Label_0012.png

Şekil 2.37. Yüz resmimizin %40'ının bilgisayar tarafından rastgele alınmasıyla oluşturulmuş kırk adet resim içeren "pxdsTest" resim kümesinin çalışma uzayı çıktısı

Tablo 2.17. Resimlerimizin boyutları 360x480 iken kırk resimlik “pxdsTest” resim kümesi için etiketlerimizin piksel sayılarının (PixelCount) ve etiketlerimizin bulunduğu resimlerin toplam piksel sayılarının (ImagePixelCount) gösterimi

Sınıf (Etiket) Adı	Etiket Piksel Sayıları (PixelCount)	Etiketlerin Bulunduğu Resimlerin Toplam Piksel Sayıları (ImagePixelCount)
'arizali_cam'	62144	6048000
'arizali_pors'	5225	1036800
'saglam_cam'	110019	6220800
'saglam_pors'	27319	2764800
'belirsiz'	35999	4838400

Tablo 2.18. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu

	'arizali_cam'	'arizali_pors'	'saglam_cam'	'saglam_pors'	'belirsiz'
'arizali_cam'	24116	79	34554	11	3384
'arizali_pors'	63	234	1693	3030	205
'saglam_cam'	34459	3	69474	0	6083
'saglam_pors'	132	563	4084	22383	157
'belirsiz'	644	109	9975	1520	23751

Tablo 2.19. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "0-1 aralığına indirgenmiş karışıklık matrisi (NormalizedConfusionMatrix)" tablosu

	'arizali_cam'	'arizali_pors'	'saglam_cam'	'saglam_pors'	'belirsiz'
'arizali_cam'	0,3881	0,0013	0,5560	0,0002	0,0545
'arizali_pors'	0,0121	0,0448	0,3240	0,5799	0,0392
'saglam_cam'	0,3132	0,0000	0,6315	0,0000	0,0553
'saglam_pors'	0,0048	0,0206	0,1495	0,8193	0,0057
'belirsiz'	0,0179	0,0030	0,2771	0,0422	0,6598

Tablo 2.20. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "veri kümesi ölçümleri (DataSetMetrics)" tablosu

Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
0,5814	0,5087	0,0218

Tablo 2.21. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu

	Doğruluk (Accuracy)	iou (birleşimlerinin kesişimi)
'arizali_cam'	0,3881	0,0210
'arizali_pors'	0,0448	0,0002
'saglam_cam'	0,6315	0,0475
'saglam_pors'	0,8193	0,0316
'belirsiz'	0,6598	0,0087

Tablo 2.22. Kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu

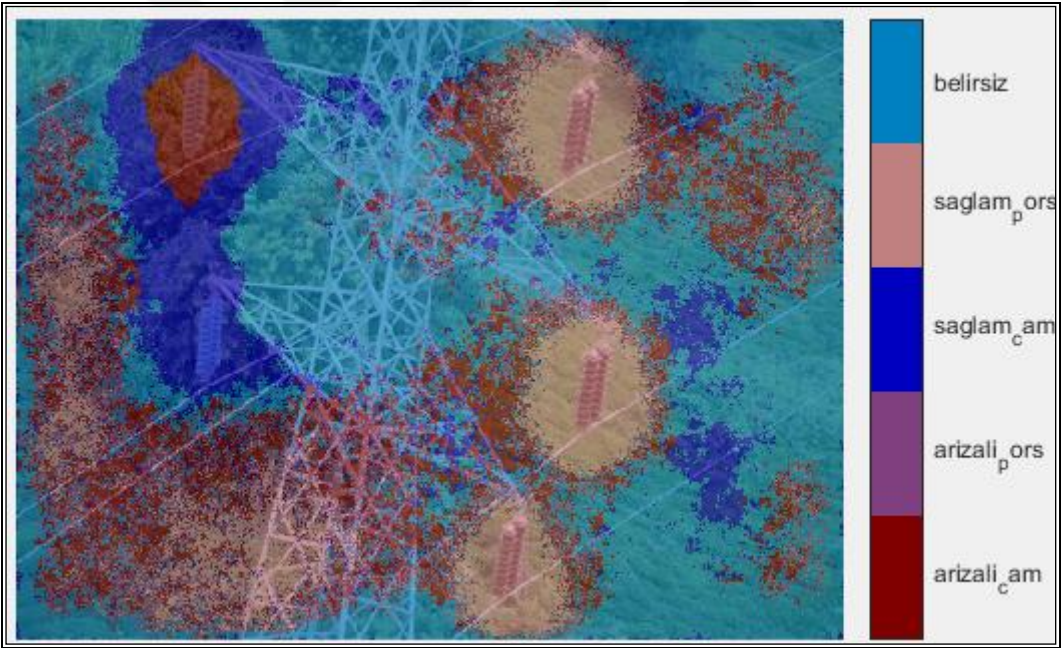
	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
1	0,6362	0,6595	0,0181
2	0,5416	0,5092	0,0152
3	0,5065	0,4340	0,0229
4	0,6615	0,7973	0,0193
5	0,8493	0,7500	0,0162
6	0,8764	0,7292	0,0220
7	0,4736	0,5209	0,0163
8	0,6804	0,6213	0,0350
9	0,7382	0,5563	0,0151
10	0,6361	0,4205	0,0054
11	0,6042	0,7074	0,0182
12	0,5960	0,4544	0,0113
13	0,5324	0,5487	0,0152
14	0,4837	0,4858	0,0270
15	0,7328	0,5924	0,0146
16	0,5951	0,4872	0,0131
17	0,5336	0,3973	0,0317
18	0,5848	0,4731	0,0173

Tablo 2.22'nin devamı

19	0,4716	0,6869	0,0325
20	0,6644	0,5445	0,0314
21	0,8711	0,6667	0,0205
22	0,6000	0,6774	0,0111
23	0,3657	0,5452	0,0178
24	0,7338	0,6488	0,0366
25	0,6007	0,7072	0,0168
26	0,6529	0,5078	0,0144
27	0,7179	0,7121	0,0187
28	0,9342	0,9161	0,0270
29	0,1726	0,1657	0,0088
30	0,7945	0,8741	0,0289
31	0,2738	0,2590	0,0055
32	0,6860	0,4768	0,0090
33	0,0856	0,0688	0,0019
34	0,9988	0,9988	0,0411
35	0,4850	0,3800	0,0152
36	0,6753	0,6903	0,0231
37	0,5933	0,5745	0,0229
38	0,8522	0,8647	0,0289
39	0,8098	0,8262	0,0244
40	0,5627	0,5465	0,0140



(a)

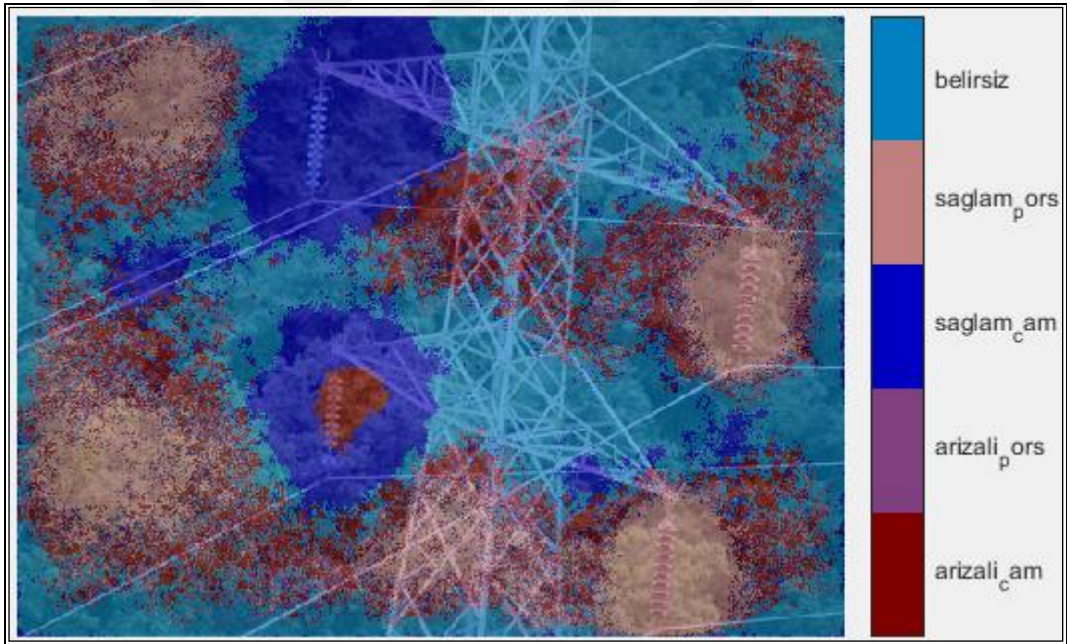


(b)

Şekil 2.38. Kırk resimlik “imdsTest” resim kümesinin 34'üncü resmi (100 resmimizin 13'üncü resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

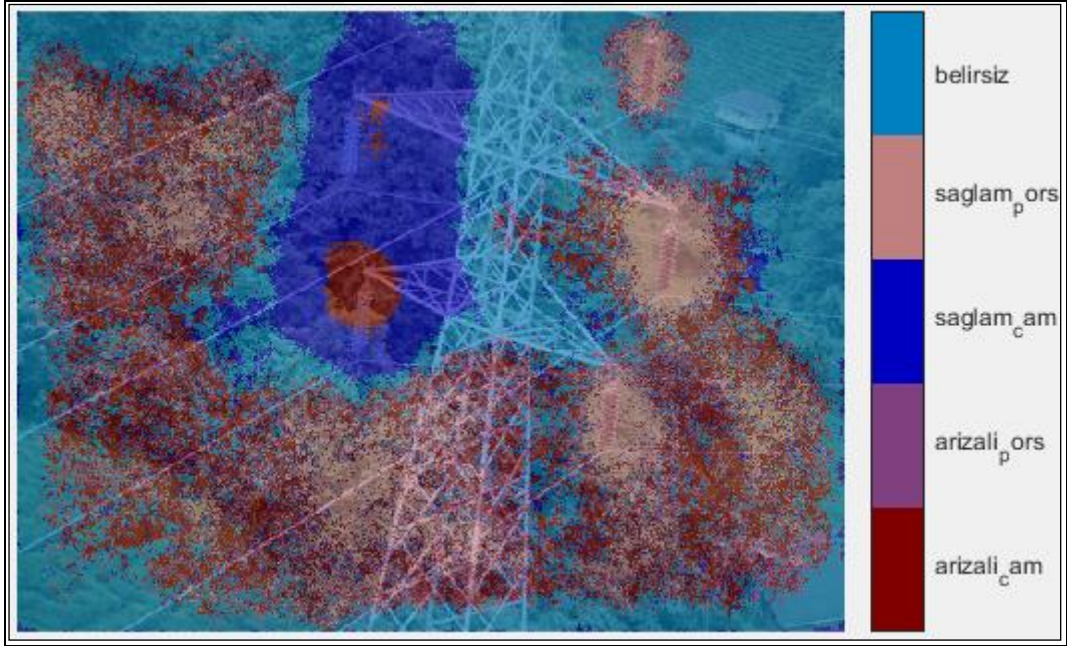


(b)

Şekil 2.39. Kırk resimlik “imdsTest” resim kümesinin 28’inci resmi (100 resmimizin 18’inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

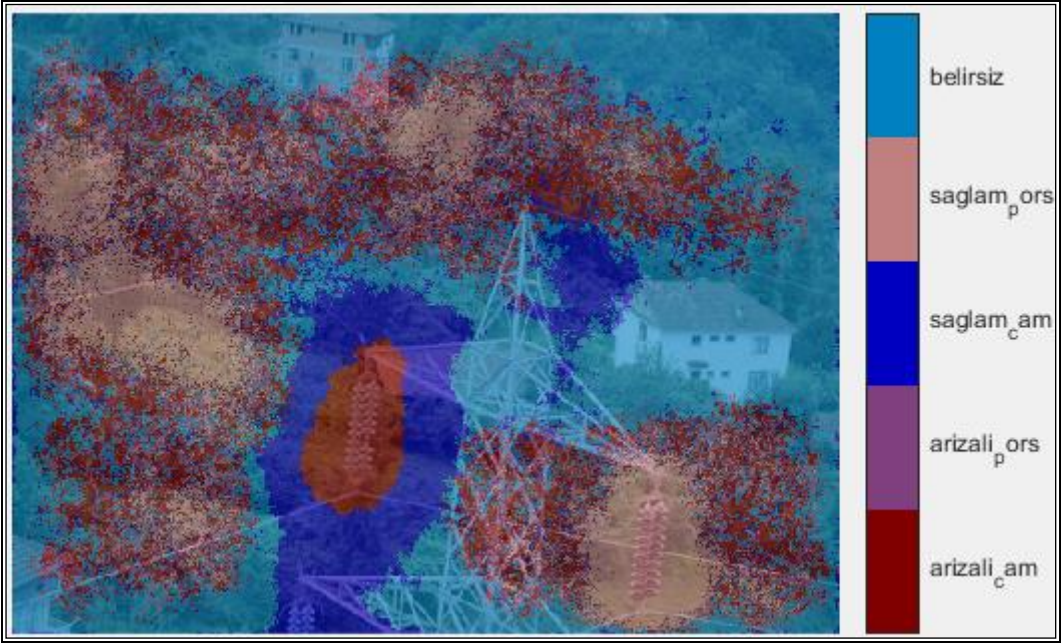


(b)

Şekil 2.40. Kırk resimlik “imdsTest” resim kümesinin 6'ncı resmi (100 resmimizin 31'inci resmi) için, (a) Eğitim öncesi, bizim etiklediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiklediği resim



(a)

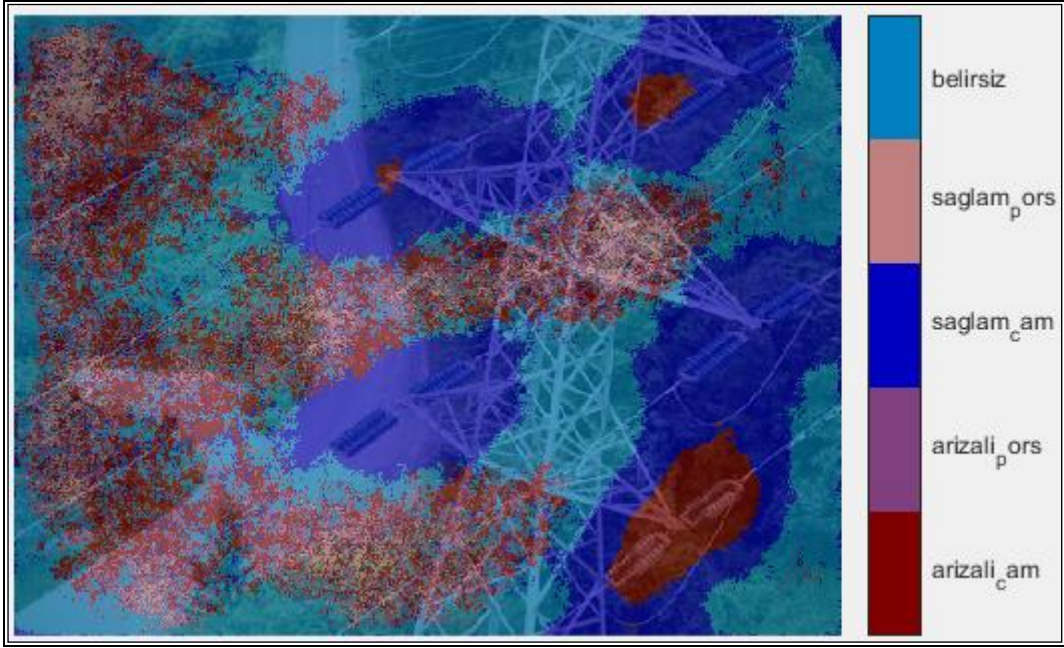


(b)

Şekil 2.41. Kırk resimlik “imdsTest” resim kümesinin 21'inci resmi (100 resmimizin 36'ncı resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

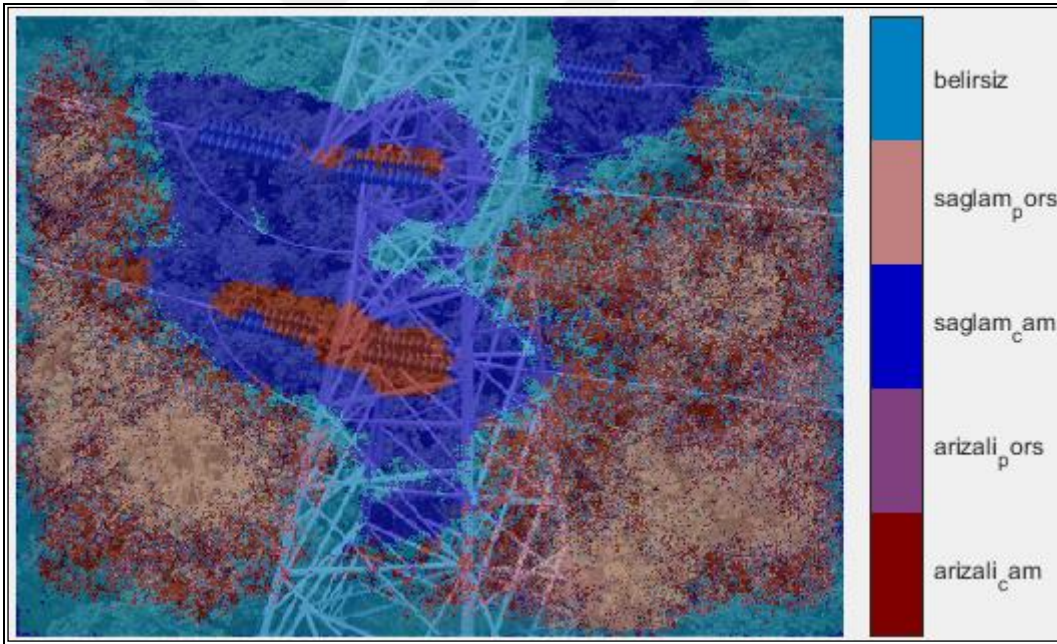


(b)

Şekil 2.42. Kırk resimlik “imdsTest” resim kümesinin 38'inci resmi (100 resmimizin 57'nci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

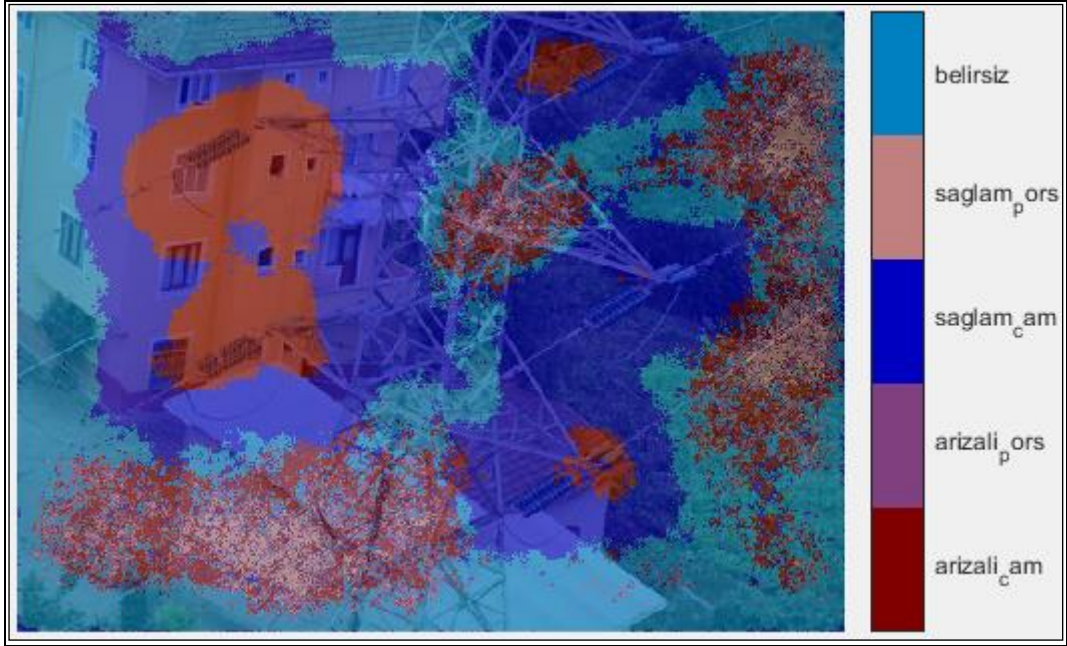


(b)

Şekil 2.43. Kırk resimlik “imdsTest” resim kümesinin 30'uncu resmi (100 resmimizin 75'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)

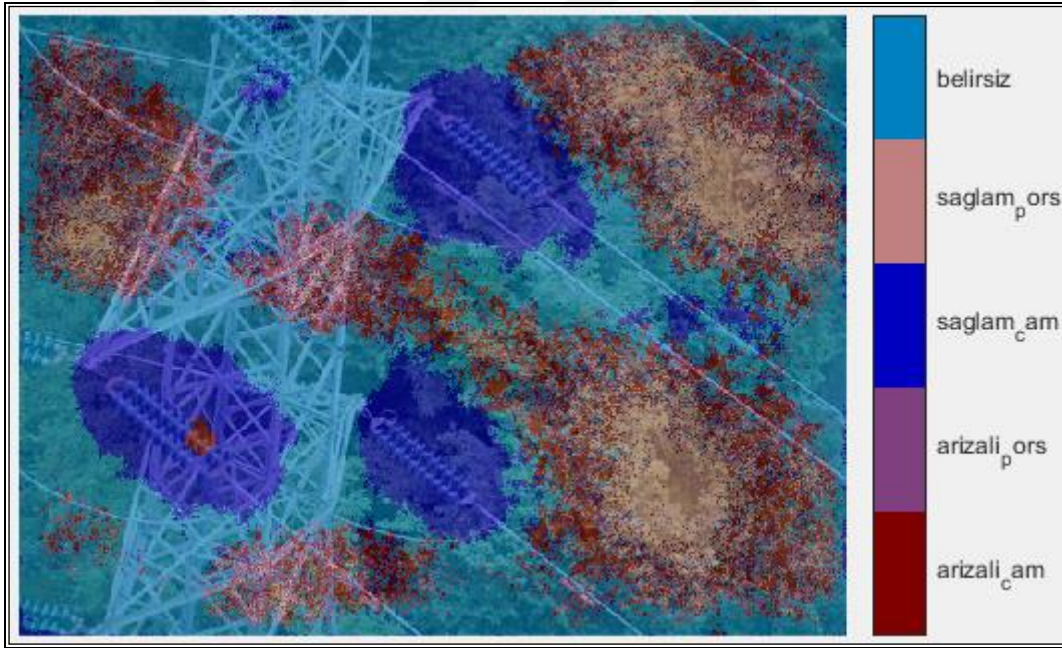


(b)

Şekil 2.44. Kırk resimlik “imdsTest” resim kümesinin 9'uncu resmi (100 resmimizin 61'inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)



(b)

Şekil 2.45. Kırk resimlik “imdsTest” resim kümesinin 24'üncü resmi (100 resmimizin 23'üncü resmi) için, (a) Eğitim öncesi, bizim etiklediğimiz resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiklediği resim

Tablo 2.23. Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.38.b, şekil 2.39.b, şekil 2.40.b, şekil 2.41.b, şekil 2.42.b, şekil 2.43.b, şekil 2.44.b ve şekil 2.45.b'de gösterilenlerinin yüzdesel doğruluk başarımlarının karşılaştırması

	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
Şekil 2.38.b	%99,88	%99,88	%4,11
Şekil 2.39.b	%93,42	%91,61	%2,70
Şekil 2.40.b	%87,64	%72,92	%2,20
Şekil 2.41.b	%87,11	%66,67	%2,05
Şekil 2.42.b	%85,22	%86,47	%2,89
Şekil 2.43.b	%79,45	%87,41	%2,89
Şekil 2.44.b	%73,82	%55,63	%1,51
Şekil 2.45.b	%73,38	%64,88	%3,66

2.10. Etiketlediğimiz Yüz Resimle Eğitilen Ağımızın Etiketlenmemiş Resimlerdeki İzolatör Sınıflarını Tespit Edip Etiketlemesi

Önceki başlıklarda doğruluk başarımlarının ölçümlenmeleri “evaluateSemanticSegmentation” komutuyla hesaplanıp matrisler halinde komut penceresinde gösterilmişti, fakat bu başlıkta bu komut çalışmayacaktır, çünkü “evaluateSemanticSegmentation” komutu, eğitim öncesi etiketlediğimiz pikseller ile eğitim sonrası bilgisayarın etiketlediği pikselleri karşılaştırarak doğruluk başarımlarını elde eder. Burada biz zaten etiketlemediğimiz resimleri bilgisayarın etiketlemesini istediğimiz için bu karşılaştırma mümkün olmayacaktır. Bu başlıkta, bu karşılaştırma yerine, etiketlediğimiz 100 resim ile eğittiğimiz ağımız yardımıyla bilgisayarımızın, hiç etiketleme yapmadığımız resimleri doğru etiketleyebilmedeki başarısı görsel olarak karşılaştırılıp yorumlanacaktır. Bunun için matlab kod bloğumuzdaki resim klasörümüzü temsil eden "images_100resim" klasörümüz yerine yeni resimlerimizi içeren "images_100yeniresim" yazılacak ve etiketlediğimiz herhangi bir resim klasörü olmadığı için kod bloğumuzdan "labels_100resim" ve bunu içeren kısımlar silinerek yeni kod bloğumuz aşağıdaki gibi oluşturulacaktır.

```

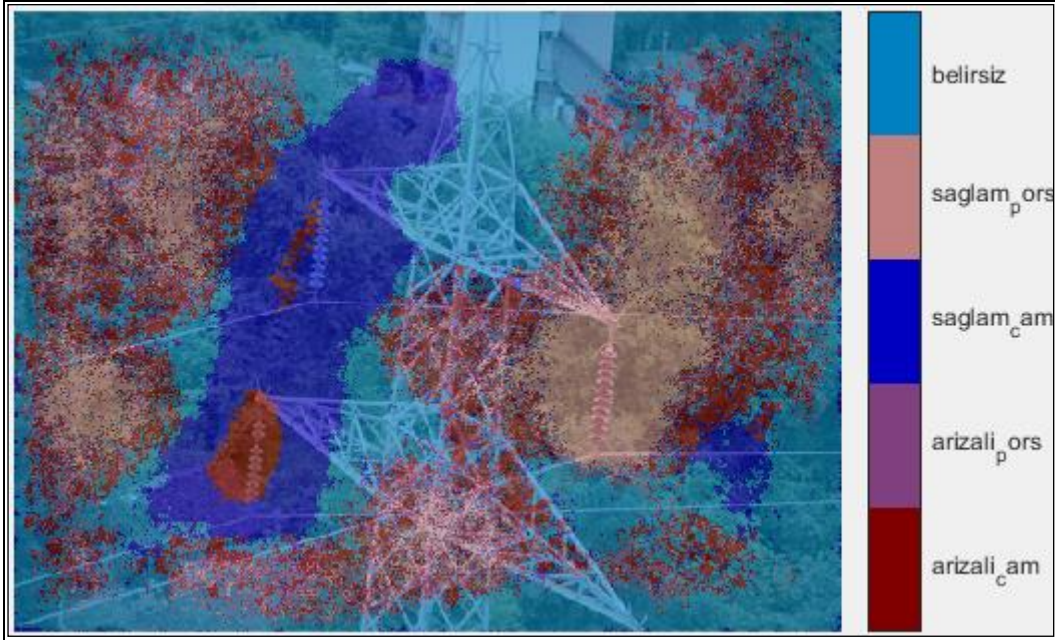
clc, clear, close all;
vgg16();
pretrainedFolder = fullfile(tempdir,'pretraineds','pretrainedSegNet');
pretrainedSegNet = fullfile(pretrainedFolder,'net.mat');
outputFolder = fullfile(tempdir,'images_labels');
imgDir = fullfile(outputFolder,'images_100yeniyesim');
imds = imageDatastore(imgDir);
classes = [
    "arizali_cam"
    "arizali_pors"
    "saglam_cam"
    "saglam_pors"
    "belirsiz"
];
cmap = camvidColorMap;
imageFolder = fullfile(outputFolder,'images_100yeniyesimResized',filesep);
imds = resizeCamVidImages(imds,imageFolder);
data = load(pretrainedSegNet);
net = data.net;
I = readimage(imds,1);
figure('Name','I')
imshow(I)
C = semanticseg(I, net);
B = labeloverlay(I,C,'Colormap',cmap,'Transparency',0.4);
figure('Name','B3=...(imds,1)')
imshow(B)
pixelLabelColorbar(cmap, classes);

```

Üstteki kod bloğunda, “I = readimage(imds,1)” satırındaki “1” rakamı yerine 1 ile 100 arasında istenilen bir sayı girilip resim klasörümüzün o sayıya denk gelen sıralamadaki resmi çağrılır, ardından eğitilmiş ağımız yardımıyla çağrılan bu resmin pikselleri bilgisayar tarafından etiketlenir. Bu şekilde oluşan resimlerin bazıları aşağıda incelenmiştir.



(a)



(b)

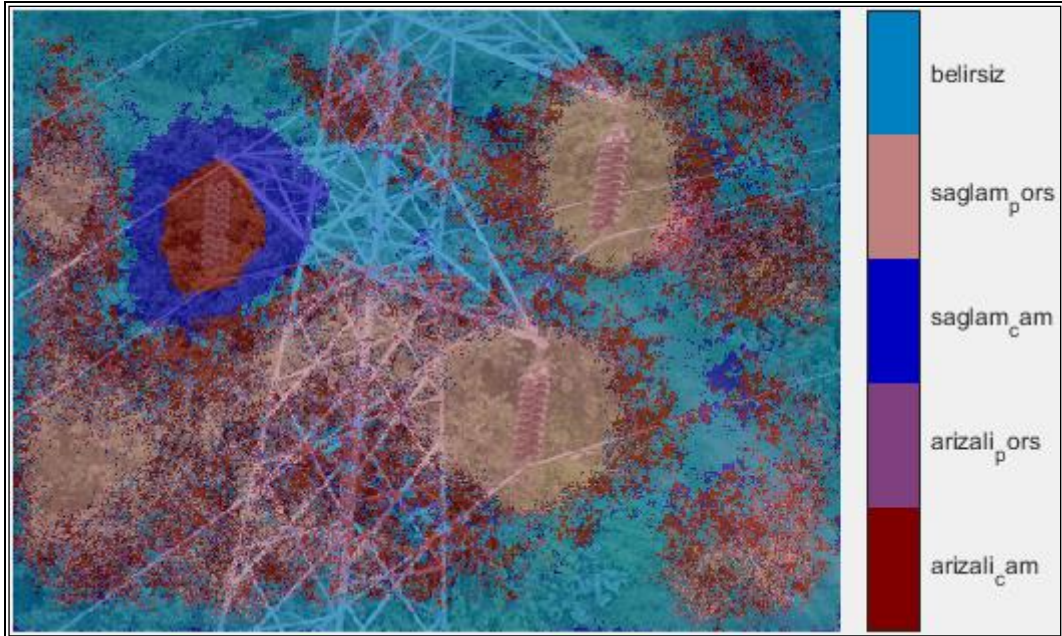
Şekil 2.46. Etiketlenmemiş 100 yeni resimden 48'inci resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Şekil 2.46'daki resmi incelediğimizde, sağdaki sağlam porselen izolatörün doğru tespit edildiği görülmektedir. Soldaki iki izolatörün ise göz ardı edilebilecek sayıda

pikseli kırmızı yerine maviye; mavi yerine kırmızıya boyanmıştır. Bu da yaklaşık 3/3 oranında yani %100'e yakın bir başarı sağlandığı anlamına gelir.



(a)



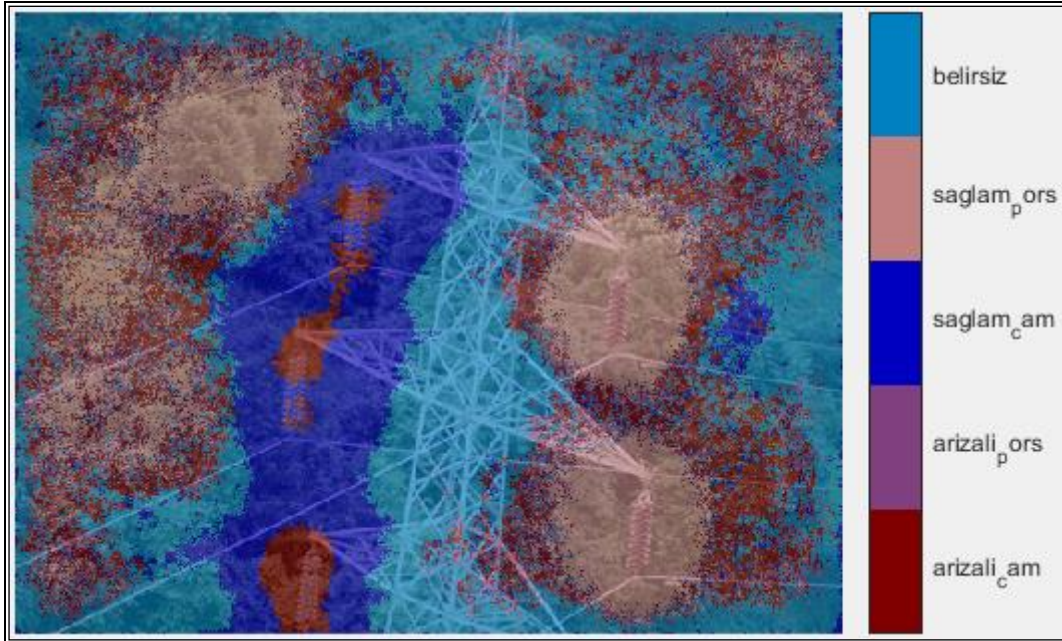
(b)

Şekil 2.47. Etiketlenmemiş 100 yeni resimden 96'ncı resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Şekil 2.47'deki resmi incelediğimizde, üç izolatörün üçünün de doğru tespit edildiği görülmektedir. Bu da 3/3 oranında yani %100 başarı sağlandığı anlamına gelir.



(a)



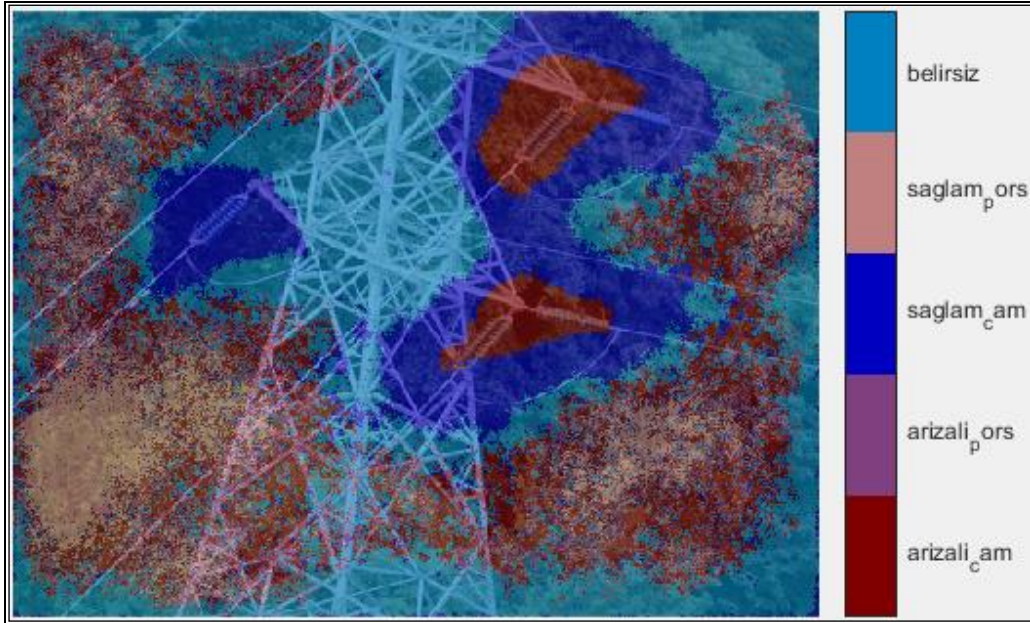
(b)

Şekil 2.48. Etiketlenmemiş 100 yeni resmin 44'üncü resmi için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Şekil 2.48'deki resmi incelediğimizde, beş izolatörün üçünün doğru tespit edildiği görülmektedir. Diğer ikisinin ise (sol üstteki iki izolatörün) tamamen maviye boyanması gerekirken yarıya yakın kısımlarının kırmızıya boyandığı görülmektedir. Bu da yaklaşık 4/5 oranında yani %80 başarı sağlandığı anlamına gelir.



(a)



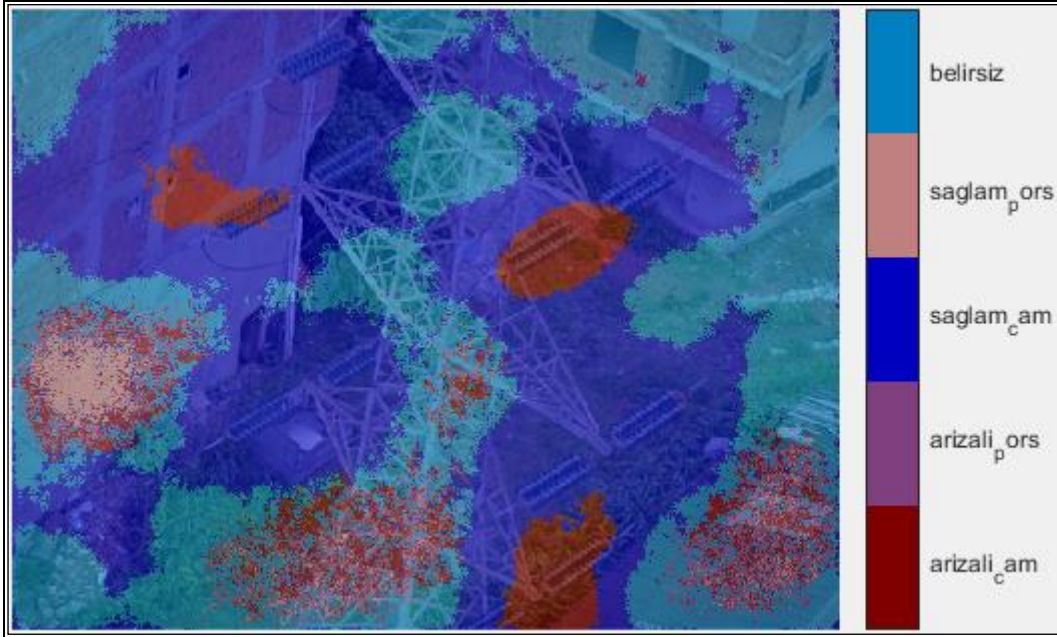
(b)

Şekil 2.49. Etiketlenmemiş 100 yeni resimden 60'ıncı resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Şekil 2.49'daki resmi incelediğimizde; resmin sağında 4, solunda 3 izolatör olduğu görülmektedir. Sağ taraftaki 4 izolatörün 3'ünün pikselleri %100'e yakın bir doğrulukla doğru tespit edilmiş fakat sağ üstteki izolatörün tamamen maviye boyanması gerekirken üçte birlik kısmı kırmızıya boyanarak üçte birlik bir hata oluşmuştur. Sol taraftaki 3 izolatörün 2'si tamamen doğru tespit edilmişken direğin arkasında kalan izolatörün tamamen kırmızıya boyanması gerekirken turkuaza boyandığı görülmektedir. Esasında eğitimlerimize bakıldığında direk arkasında kalan izolatörlerin büyük bir çoğunluğunu turkuaz olarak etiketlemiştik, bu yüzden bilgisayarın burada direk arkasındaki izolatörü turkuaza boyaması eğitim açısından doğru bir karar olsa da resme yakından bakıldığında direk arkasındaki izolatörün bir dilimin kopmuş olduğu ve arızalı olarak kırmızı renge boyanması gerektiği de bir gerçektir, işte bu yüzden bu izolatör hatalı olarak kabul edilmiştir. Böylece 7 izolatörün 5 tanesinin tamamen doğru, 1 tanesinin 2/3 oranında doğru, diğerinin (direk arkasındaki) ise yanlış tespit edilmiş olduğu görülür. Bu da yaklaşık 5,66/7 oranında yani yaklaşık %80 başarı sağlandığı anlamına gelir.



(a)



(b)

Şekil 2.50. Etiketlenmemiş 100 yeni resimden 22'nci resim için, (a) Etiketlenmemiş resim, (b) Yüz resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Şekil 2.50'deki resmi incelediğimizde; resmin sağında 6, solunda 6 izolator olduğu görülmektedir. Sağ taraftaki tamamı maviye boyanan 3 izolator doğru tespit edilmişken kırmızıya boyanan 2 izolatörden alt tarafta olanının (en alttakinin bir üstündeki izolator)

tamamen kırmızıya boyanması gerekirken yaklaşık 1/4'lük kısmı yanlış olarak maviye boyanmıştır. En alttaki izolatörün ise tamamen turkuaza boyanması gerekirken bir kısmının maviye bir kısmının da kırmızıya boyanmış olduğu görülür. Sol taraftaki 6 izolatörden tamamı maviye boyanan 4 izolatörün doğru tespit edildiği, yaklaşık 3/4'ü kırmızıya boyanan izolatörün ise tamamen maviye boyanması gerektiği, en üstteki izolatörün ise turkuaza boyanarak doğru tespit edildiği görülür. Böylece sağdaki 6 izolatörün 3'ünün doğru, bir tanesinin 3/4'ünün doğru tespit edildiği görülmüşken soldaki 6 izolatörün ise 5'inin doğru, bir tanesinin 1/4'ünün doğru tespit edildiği görülmüştür. Bu da yaklaşık 9/12 oranında yani %75 başarı sağlandığı anlamına gelir.

2.11. Arızalı ve Sağlam İzolatörleri Tespit Etmedeki Doğruluk Başarım Oranının Yükseltmesine Yönelik Eğitim Aktarımı (Transfer Learning) Yönteminin Kullanılması

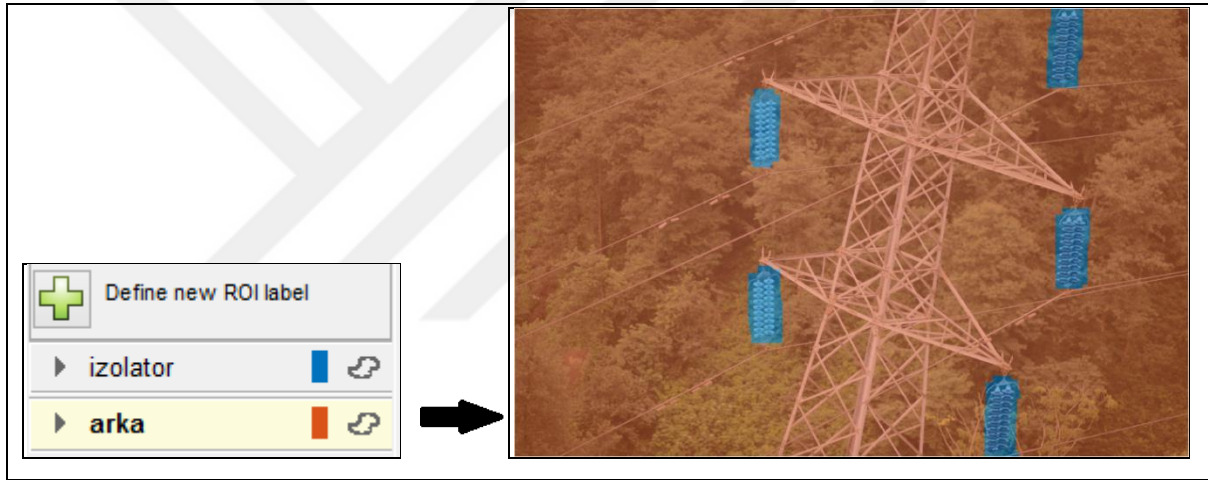
Önceki başlıklarda özet olarak belirli sayıda resim (önce 40 resim, ardından 100 resim) etiketlenerek oluşturulan eğitilmiş ağ yardımıyla bilgisayarın, test kümesindeki resimleri doğru etiketlemedeki başarı oranı ölçülmüş ve kırk resim için %53, yüz resim için %58 başarı oranları tespit edilmiştir. Bu başlıkta ise bu doğruluk başarımlarını artırmak için yapılan çalışmalar aşağıda maddeler halinde özetlenmiştir.

1. Önce, yüz yetmiş adet resim, "image labeler" uygulamasıyla "arka plan" ve "izolatör" olmak üzere iki sınıf olarak etiketlenmiştir.
2. Ardından, iki sınıf ile etiketlenmiş bu yüz yetmiş adet resim eğitime sokulmuştur. Eğitim sonucu oluşan test kümeleri için bilgisayarın bu iki sınıfı tespit etmesindeki doğruluk başarımları bulunmuştur.
3. Sonrasında, elde edilen eğitilmiş ağımız yardımıyla etiketleme işlemine tabi tutulmayan yeni resimlerdeki izolatörler sağlam veya arızalı olmasına bakılmaksızın bilgisayar tarafından otomatik olarak tespit edilmiştir.
4. Belirlediğimiz yüz yetmiş resme "imcrop" komutu ile kesme işlemi uygulanarak resimlerin içindeki arızalı ve sağlam izolatörler iki klasöre kaydedilmiştir.
5. İki klasöre kaydedilen arızalı ve sağlam izolatörler, öğrenme transferi (transfer learning) yöntemi kullanılarak önceden eğitilmiş bir ağ olan "alexnet" ağı yardımıyla eğitime sokulmuştur. Eğitim sonucu oluşan test kümeleri için bilgisayarın arızalı ve sağlam izolatörleri tespit etmesindeki doğruluk başarımları bulunmuştur.

6. Öğrenme transferi (transfer learning) yöntemi kullanılarak oluşturulmuş eğitilmiş ağızımız yardımıyla, yukarıdaki üçüncü maddede bahsedilen etiketleme işlemine tabi tutulmayan yeni resimlerde saptanan izolatorların sağlam mı arızalı mı olduğu tespit edilmiştir.

2.11.1. Yüz Yetmiş Adet Resmin İki Sınıflı Etiketlenmesi, Eğitimi ve Eğitim Sonuçları

Aşağıdaki şekil 2.51'deki gibi yüz yetmiş adet resim "izolator" ve "arka" olmak üzere iki sınıf olarak etiketlenmiştir. Amaç, izolatorların yerlerinin tespit edilmesi olduğu için arızalı ve sağlam izolator ayrımı gözetmeksizin aynı sınıf adı ile etiketleme yapılmıştır.



Şekil 2.51. Klasördeki ilk fotoğrafımız olan şekil 2.1'deki fotoğrafın "image labeler" uygulaması yardımıyla iki sınıflı etiketlemesinin yapılmış hali

İki sınıflı etiketleme işleminin yapılmasının ardından eğitime geçilmiştir. Eğitim için önceki başlıklarda anlattığımız ana kod bloğumuzdaki "classes" değişkeni ile "camvidPixelLabelIDs" ve "camvidColorMap" fonksiyonları içindeki beş sınıf, iki sınıfa düşürülmüştür.

İki sınıf ve 170 resim ile ağız eğitilmesi; "160867,13" saniye yani yaklaşık 45 saat sürmüştür. Eğitim sonucunda test kümeleri için tüm resim sayısının %40'ı oranında resim seçileceği için "imdsTest" ve "pxdsTest" resim kümelerine 68'er adet resim seçilmiştir.

Tablo 2.24. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "karışıklık matrisi (ConfusionMatrix)" tablosu

	'izolator'	'arka'
'izolator'	508070	28598
'arka'	322349	10871925

Tablo 2.25. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan 0-1 aralığına indirgenmiş karışıklık matrisi (normalized confusion matrix) tablosu

	'izolator'	'arka'
'izolator'	0,9467	0,0533
'arka'	0,0288	0,9712

Tablo 2.26. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "veri kümesi ölçümleri (DataSetMetrics)" tablosu

Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
0,9701	0,9590	0,7793

Tablo 2.27. Altmış sekizer resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz yetmiş resimle eğitilmiş ağ kullanılarak oluşturulan "sınıf ölçümleri (ClassMetrics)" tablosu

	Doğruluk (Accuracy)	IoU (birleşimlerinin kesişimi)
'izolator'	0,9467	0,5915
'arka'	0,9712	0,9671

Genel doğruluk (global accuracy) değerinin %97 ve "MeanIoU" değerinin %77,9 olduğu tablo 2.26'da; sınıf ölçümleri (class metrics) değerleri için "izolator" sınıfı değerinin %94,67 ve "arka" sınıfı değerinin de %97,12 olduğu tablo 2.27'de görülmektedir. Bu değerler, izolatörlerin yerlerinin tespitinde gayet başarılı bir sonuç elde edildiğini göstermektedir.

Test kümesindeki her resim için izolatörlerin yerlerinin tespit edilmesindeki başarı oranlarını gösteren genel doğruluk (global accuracy) değerlerinin %93 ile %99 aralığında

olduđu; bilgisayarın, yanlış tespit sonucu izolatör olmayan yerlerdeki bazı pikselleri izolatör olarak etiketlemesi sonucu "MeanIoU" değerlerinin %58 ile %86 aralığında deđiřtiđi tablo 2.28'deki resim ölçümleri (image metrics) tablosunda gösterilmiřtir.

Tablo 2.28. Altmış sekizer resimlik "imdsTest" ve "pxdsTest" resim kümeleri için yüz yetmiş resimle eğitilmiş ađ kullanılarak oluşturulan "resim ölçümleri (ImageMetrics)" tablosu

	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
1	0,9576	0,9642	0,7661
2	0,9698	0,9765	0,8245
3	0,9845	0,9720	0,7653
4	0,9718	0,9634	0,8515
5	0,9658	0,9629	0,8075
6	0,9348	0,9480	0,7529
7	0,9659	0,9433	0,7575
8	0,9717	0,9852	0,7655
9	0,9618	0,9687	0,7433
10	0,9874	0,9805	0,8468
11	0,9841	0,9876	0,7721
12	0,9790	0,9876	0,7335
13	0,9592	0,9753	0,7224
14	0,9895	0,9695	0,8260
15	0,9811	0,9875	0,7234
16	0,9634	0,9277	0,7845
17	0,9499	0,9549	0,7621
18	0,9666	0,9811	0,7473
19	0,9709	0,9320	0,7506
20	0,9578	0,9758	0,7415
21	0,9763	0,9747	0,8203
22	0,9300	0,9449	0,7609
23	0,9724	0,9126	0,7556
24	0,9515	0,9449	0,7864
25	0,9776	0,8829	0,6960
26	0,9850	0,6440	0,5842
27	0,9811	0,9879	0,7413
28	0,9895	0,9688	0,8693
29	0,9701	0,9824	0,7767
30	0,9532	0,9674	0,8160
31	0,9809	0,9817	0,7739
32	0,9484	0,9491	0,7688
33	0,9301	0,8895	0,7441
34	0,9804	0,9819	0,8261
35	0,9805	0,8836	0,7110
36	0,9840	0,9882	0,7530

Tablo 2.28'in devamı

37	0,9561	0,9749	0,7759
38	0,9763	0,9878	0,7442
39	0,9462	0,9518	0,7675
40	0,9606	0,9675	0,8234
41	0,9818	0,9675	0,7599
42	0,9870	0,9719	0,7617
43	0,9335	0,9572	0,7756
44	0,9717	0,9816	0,7884
45	0,9912	0,9765	0,8025
46	0,9861	0,9431	0,7553
47	0,9761	0,9816	0,7326
48	0,9773	0,9882	0,7806
49	0,9560	0,9721	0,7804
50	0,9704	0,9628	0,7780
51	0,9833	0,9410	0,8123
52	0,9819	0,9885	0,7358
53	0,9544	0,9506	0,7926
54	0,9884	0,9014	0,8095
55	0,9671	0,7556	0,6188
56	0,9867	0,9731	0,8061
57	0,9777	0,9787	0,7541
58	0,9466	0,8643	0,7193
59	0,9812	0,8894	0,6096
60	0,9824	0,9843	0,8340
61	0,9657	0,9800	0,7763
62	0,9764	0,9786	0,8328
63	0,9627	0,9397	0,7998
64	0,9641	0,9700	0,8022
65	0,9804	0,9883	0,7518
66	0,9819	0,9455	0,7774
67	0,9696	0,9827	0,7743
68	0,9815	0,9764	0,8090

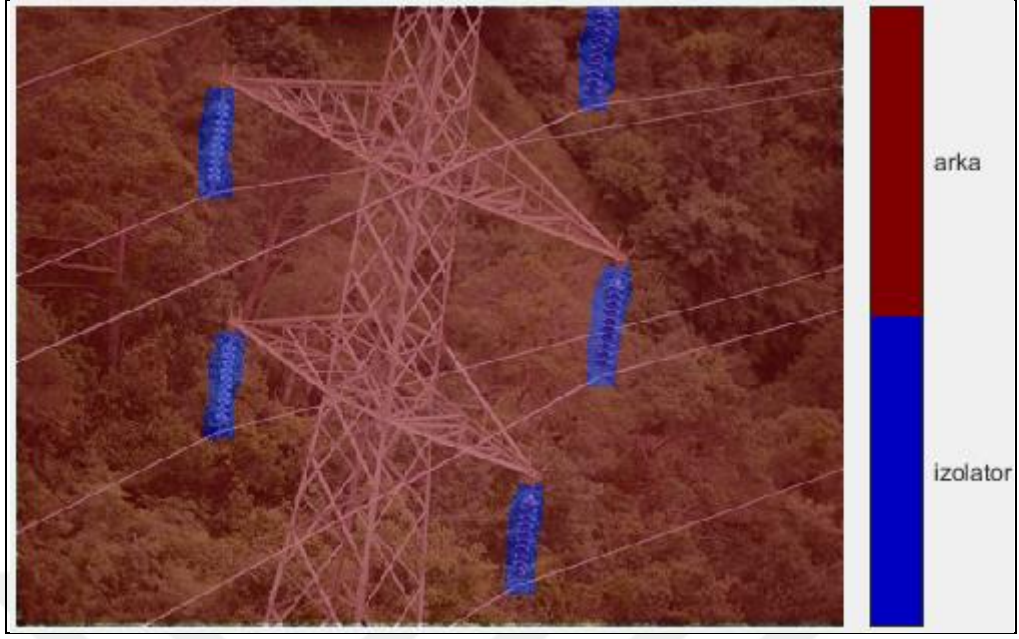


(a)

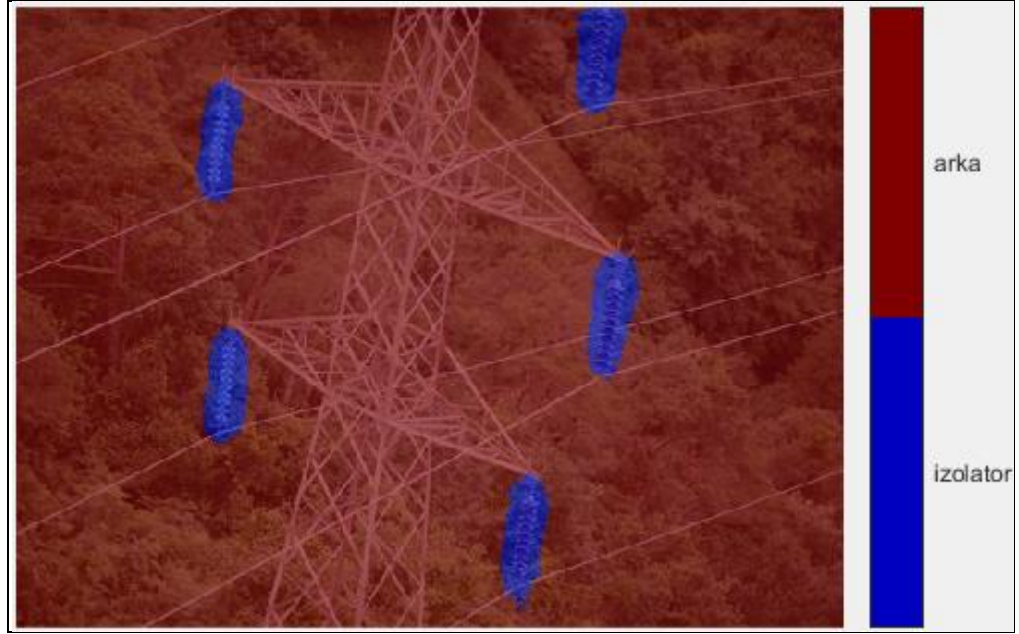


(b)

Şekil 2.52. Altmış sekiz resimlik “imdsTest” resim kümesinin 45’inci resmi (170 resmimizin 158’inci resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

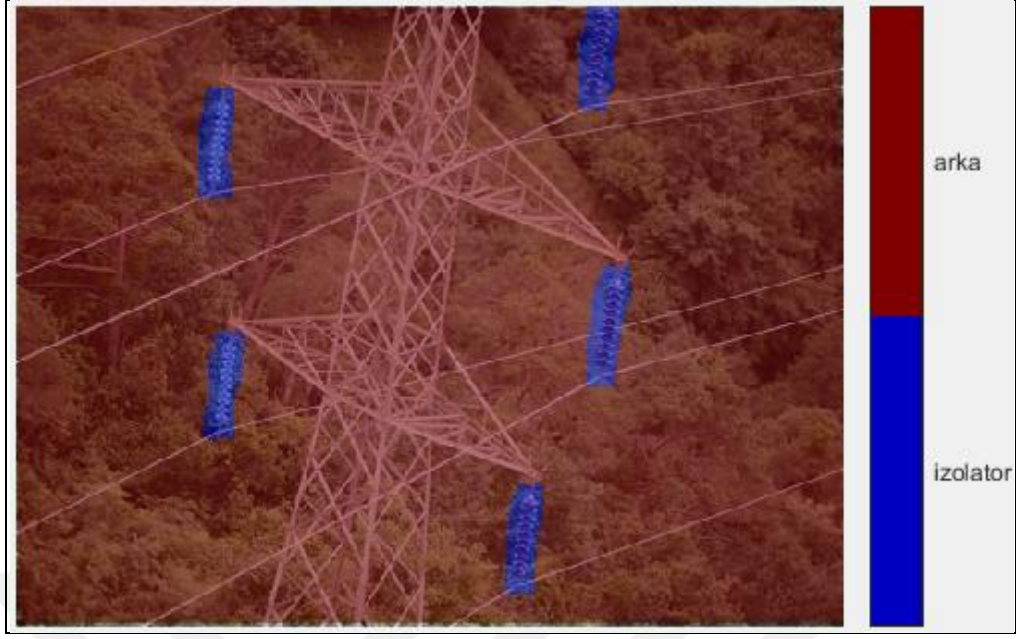


(a)

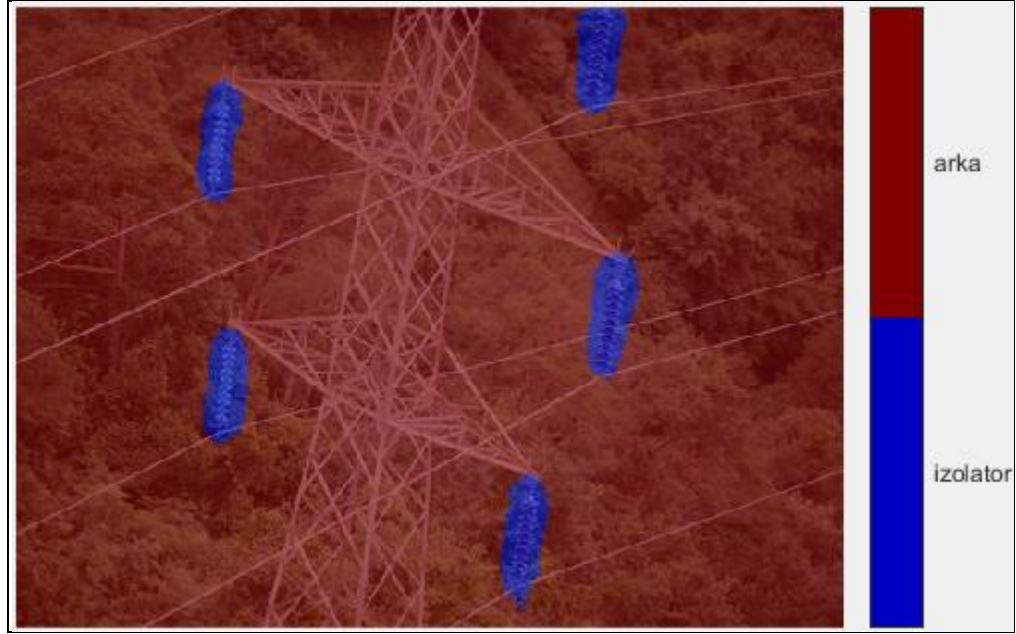


(b)

Şekil 2.53. Altmış sekiz resimlik “imdsTest” resim kümesinin 28'inci resmi (170 resmimizin 19'uncu resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

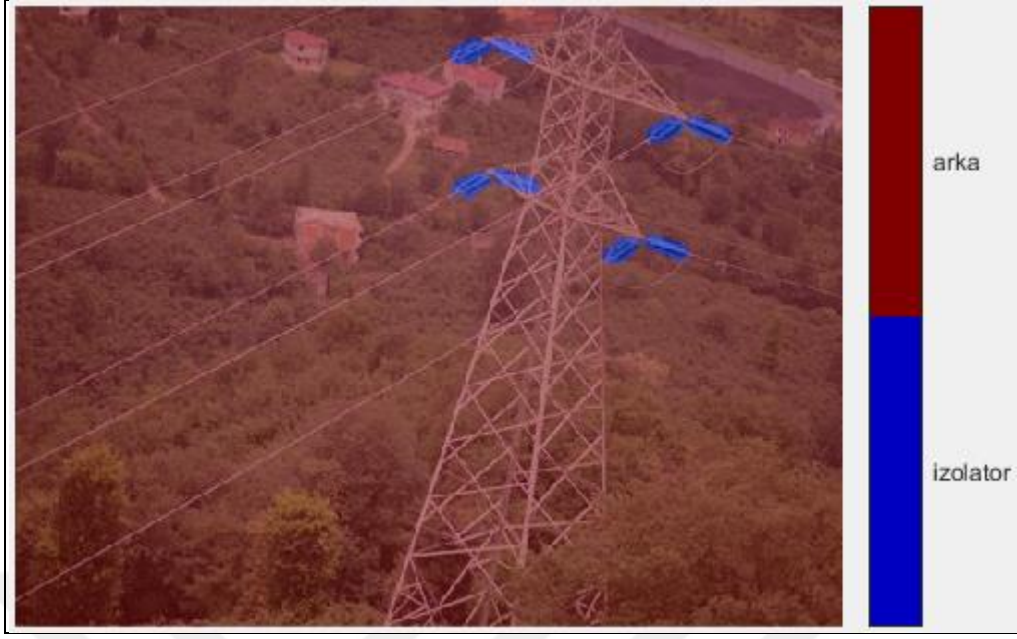


(a)



(b)

Şekil 2.54. Altmış sekiz resimlik “imdsTest” resim kümesinin 22'nci resmi (170 resmimizin 87'nci resmi) için, (a) Eğitim öncesi, bizim etiklediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiklediği resim



(a)



(b)

Şekil 2.55. Altmış sekiz resimlik “imdsTest” resim kümesinin 26'ncı resmi (170 resmimizin 94'üncü resmi) için, (a) Eğitim öncesi, bizim etiketlediğimiz resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

Tablo 2.29. Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resimlerden; şekil 2.52.b, şekil 2.53.b, şekil 2.54.b ve şekil 2.55.b'de gösterilenlerinin yüzdesel doğruluk başarımlarını karşılaştırması

	Genel Doğruluk (Global Accuracy)	Ortalama Doğruluk (Mean Accuracy)	Ortalama IoU (Mean IoU)
Şekil 2.52.b	%99,12 (En iyi sonuç)	%97,65	%80,25
Şekil 2.53.b	%98,95	%96,88	%86,93 (En iyi sonuç)
Şekil 2.54.b	%93,00 (En kötü sonuç)	%94,49	%76,09
Şekil 2.55.b	%98,50	%64,40	%58,42 (En kötü sonuç)

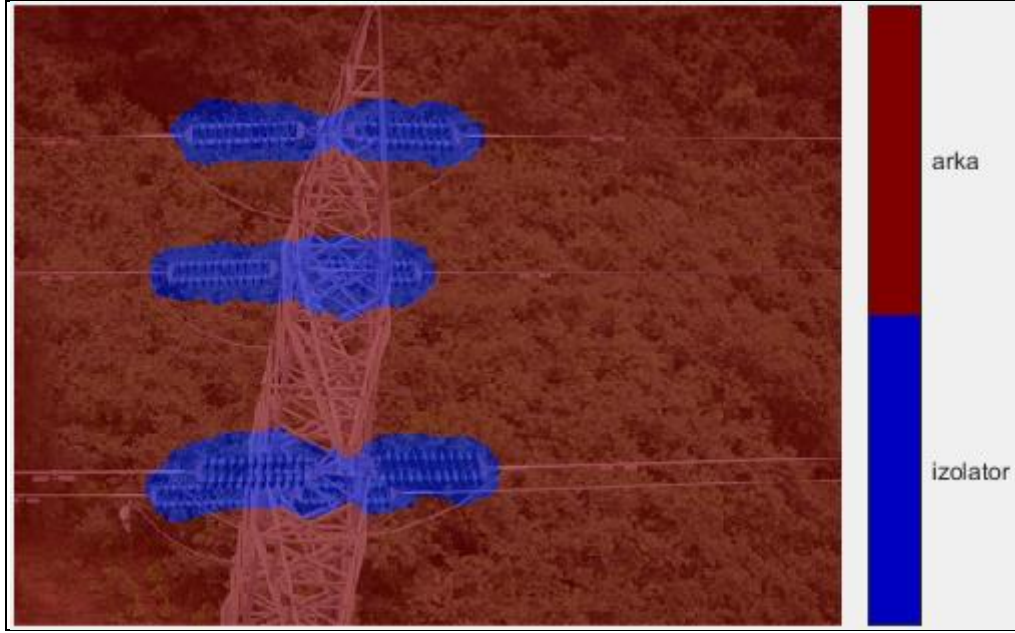
2.11.2. Etiketlediğimiz Yüz Yetmiş Resim ile Eğitilmiş Ağımızın Etiketlenmemiş Resimlerdeki İzolatörlerin Yerlerini Tespit Edip Etiketlemesi

Bölüm 2.10'da yazdığımız kod bloğu kullanılmıştır. Bu kod bloğundaki "classes" değişkeni içindeki sınıf sayısı ikiye düşürülerek "izolator" ve "arka" sınıfları oluşturulmuş, verileri gireceğimiz "imds" değişkenine etiketlenmemiş resimlerimizin bulunduğu klasör adı atanmış, "camvidColorMap" fonksiyonu içindeki sınıf sayısı ikiye düşürülmüştür. Bu değişiklikler yapıldıktan sonra kod bloğumuz çalıştırılarak iki sınıf ve 170 resimle eğitilmiş ağımız yardımıyla etiketlenmemiş resimlerimizdeki izolator yerleri bilgisayar tarafından tespit edilmiştir. Yerleri tespit edilen izolatorü gösteren resimlerden bir kaç tanesi aşağıda gösterilmiştir. Hatırlatılması gereken bir nokta da, bölüm 2.10'da bahsedildiği gibi etiketlemediğimiz resimleri bilgisayarın etiketlemesini istediğimiz için komutların çalıştırılması sonucu herhangi bir yüzdelik başarımlarının karşımıza çıkmayacağıdır. Bu yüzden resimlerdeki izolatorlerin yerlerinin doğru tespit edilip edilmediği görsel olarak karşılaştırılıp yorumlanmıştır.

Aşağıdaki şekil 2.56, şekil 2.57 ve şekil 2.58'deki resimleri incelediğimizde, İzolatörlerin yerlerinin doğru tespit edildiği görülmektedir.



(a)



(b)

Şekil 2.56. Etiketlenmemiş 70 yeni resmin 67'nci resmi için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)



(b)

Şekil 2.57. Etiketlenmemiş 70 yeni resmin 21'inci resmi için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim



(a)



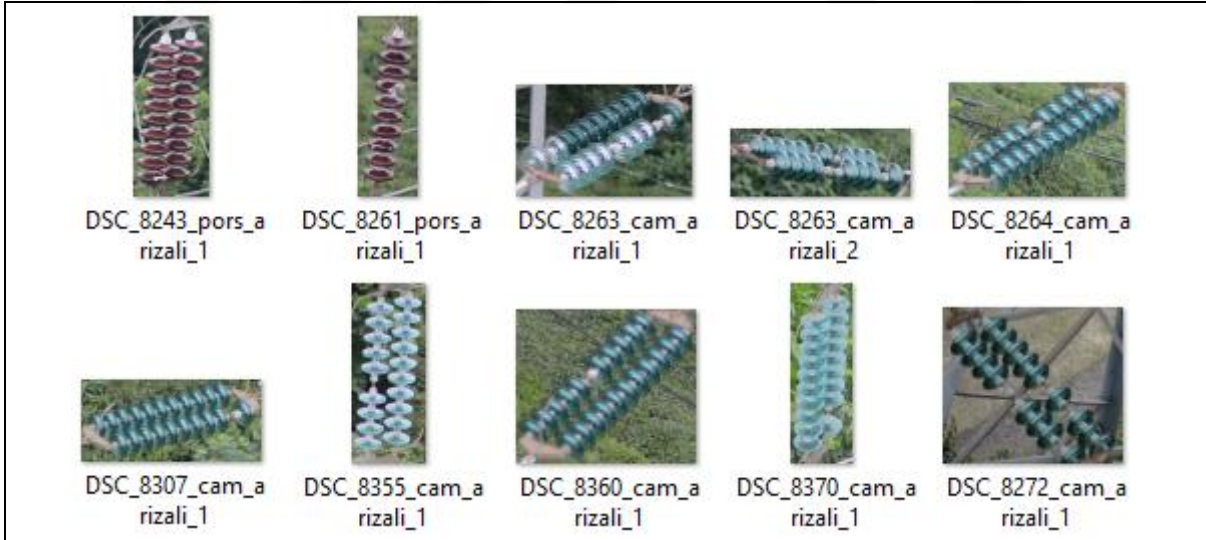
(b)

Şekil 2.58. Etiketlenmemiş 70 yeni resimden 14'üncü resim için, (a) Etiketlenmemiş resim, (b) Yüz yetmiş resim ile eğitilmiş ağ kullanılarak bilgisayarın etiketlediği resim

2.11.3. Yüz Yetmiş Adet Resimdeki Arızalı ve Sağlam İzolatörlerin "imcrop" Komutu ile Kesilerek Ayrı Klasörlere Kaydedilmesi

Bir önceki bölümde yerleri tespit edilen izolatörlere bu bölümde "imcrop" komutu ile kesme işlemi uygulanmış ve elde edilen izolatörler "arızalı" ve "sağlam" olmak üzere iki ayrı klasöre kaydedilmiştir. Yüz yetmiş adet resimden tespit edilen 783 adet izolatör, 610 adeti sağlam ve 173 adeti arızalı olmak üzere ayrı klasörlere kaydedilmiştir. Kullandığımız kod bloğunun bir kısmı aşağıdaki gibidir.

```
im1 = imread('DSC_9050.JPG'); % resim okutulur.
im1_1 = imcrop(im1); % imcrop komutu ile kesme işlemi yapılır.
a1 = imcrop(im1,[4507.5 0.5 1135 639]); % İzolatörlerin koordinatları kopyalanır.
imwrite(a1,'C:\Users\Hp\...\sağlam_izolator\DSC_9050_saglam_1.png'); % Koordinatları
% kopyalanan izolatörler arızalı ve sağlam olmak üzere ayrı klasörlere kaydedilir.
a2 = imcrop(im1,[3845.5 2167.5 1132 758]);
imwrite(a2,'C:\Users\Hp\...\arizali_izolator\DSC_9050_sarizali_1.png');
```



Şekil 2.59. Kesme işlemi sonucu oluşan arızalı izolatörlerden bazıları



Şekil 2.60. Kesme işlemi sonucu oluşan sağlam izolatörlerden bazıları

2.11.4. Önceden Eğitilmiş Bir Ağ Olan "Alexnet" Ağı Yardımıyla Öğrenme Transferi (Transfer Learning) Yöntemi Kullanılarak İzolatörlerin Arızalı ve Sağlam Olarak Tespiti

Bir önceki bölümde iki ayrı klasöre ayrılan arızalı ve sağlam izolatörler bu bölümde eğitime sokulmuştur. Eğitim için önceden eğitilmiş bir ağ olan "alexnet" ağına "arızalı_izolator" ve "sağlam_izolator" sınıflarımız için eğitim aktarımı (transfer learning) yapılmıştır. 25 katmandan (layers) oluşan "alexnet" ağının son üç katmanı, eğitim aktarımı (transfer learning) için kullanılmıştır. "alexnet" ağı 227x227 boyutlarındaki resimleri desteklediği için eğitime başlamadan önce izolatör resimlerimiz "imresize" komutu ile "227x227" boyutlarına dönüştürülmüştür. Kullandığımız kod bloğu parçalara ayrılarak aşağıda anlatılmıştır [19].

```
images = imageDatastore('izolatorler',...
    'IncludeSubfolders',true,...
    'LabelSource','foldernames');
tbl = countEachLabel(images)
minSetCount = min(tbl{:,2});
resimler2 = splitEachLabel(resimler, minSetCount, 'randomize');
tbl2=countEachLabel(images2)
```

Üstteki kod bloğu parçasında, "imageDatastore" komutu yardımıyla "images" değişkenine "izolatorler" klasörü ile birlikte içerisindeki "arızalı_izolator" ve

“saglam_izolator” klasörleri atanır. “countEachLabel” komutuyla “images” değişkeni içinde bulunan “arizali_izolator” ve “saglam_izolator” klasörleri içerisindeki izolatörler sayılır. Sayılan bu izolatörler tablo halinde “tbl” değişkenine atanır. Şekil 2.61’de görüldüğü gibi arızalı izolatörlerin sayısı 173, sağlam izolatörlerin sayısı ise 610 olarak bulunur. “min” komutu yardımıyla “tbl” değişkenine atanan “arizali_izolator” ve “saglam_izolator” klasörlerinden içerdiği resim sayısı küçük olan “arizali_izolator” klasörünün izolatör sayısı olan 173 sayısı, “minSetCount” değişkenine atanır. “splitEachLabel” komutu yardımıyla “images” değişkeni içerisinde bulunan “arizali_izolator” ve “saglam_izolator” klasörlerinin izolatör sayıları “minSetCount” değişkenine eşitlenir. Böylece sağlam izolatörlerin sayısı ise 610’dan 173’e düşürülmüş olurken zaten 173 tane olan arızalı izolatörlerin sayısı da değişmemiş olur (şekil 2.61). “randomize” komutu yardımıyla 610 adet sağlam izolatör içerisinden 173 adedi rastgele seçilir. Rastgele seçilen bu 173 adet sağlam izolatör ve zaten sayısı 173 olan arızalı izolatörler “images2” değişkenine atanır. “countEachLabel” komutuyla “images2” değişkeni içinde bulunan “arizali_izolator” ve “saglam_izolator” klasörleri içerisindeki izolatörler sayılır. Sayılan bu izolatörler tablo halinde “tbl2” değişkenine atanır.

Sınıf (Etiket) Adı	Sınıf (Etiket) Sayısı		Sınıf (Etiket) Adı	Sınıf (Etiket) Sayısı
'arizali_izolator'	173	➔	'arizali_izolator'	173
'saglam_izolator'	610		'saglam_izolator'	173

Şekil 2.61. “arizali_izolator” ve “saglam_izolator” sınıf sayılarının eşitlenmesi

```
[trainingImages,validationImages] = splitEachLabel(images2,0.7,'randomized');
tbl3=countEachLabel(trainingImages)
tbl4=countEachLabel(validationImages)
```

Üstteki kod bloğu parçasında, “splitEachLabel” komutu yardımıyla “images2” değişkeni içerisinde bulunan 173’er adet arızalı ve sağlam izolatörün %70’i “randomize” komutuyla rastgele seçilerek “trainingImages” değişkenine atanırken geri kalan %30’luk kısım ise “validationImages” değişkenine atanır. “trainingImages (eğitim kümesini oluşturan resimler)” ile “validationImages (doğrulama kümesini oluşturan resimlerin)” sayısı, şekil 2.62’de gösterilmiştir.

Sınıf (Etiket) Adı	“trainingImages” yani eğitim kümesini oluşturan resimlerin sayısı	⇒	Sınıf (Etiket) Adı	“validationImages” yani doğrulama kümesini oluşturan resimlerin sayısı
'arizali_izolator'	121		'arizali_izolator'	52
'saglam_izolator'	121		'saglam_izolator'	52

Şekil 2.62. Arızalı ve sağlam izolatör klasörleri için, “trainingImages (eğitim kümesini oluşturan resimler)” ile “validationImages (doğrulama kümesini oluşturan resimler)” sayıları

```

net = alexnet;
layersTransfer = net.Layers(1:end-3);
numClasses = numel(categories(trainingImages.Labels))
layers = [
    layersTransfer

    fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)
    softmaxLayer
    classificationLayer];
miniBatchSize = 20;
numIterationsPerEpoch = floor(numel(trainingImages.Labels)/miniBatchSize);
options = trainingOptions('sgdm',...
    'MiniBatchSize',miniBatchSize,...
    'MaxEpochs',4,...
    'InitialLearnRate',1e-4,...
    'Verbose',false,...
    'Plots','training-progress',...
    'ValidationData',validationImages,...
    'ValidationFrequency',numIterationsPerEpoch);

```

Üstteki kod bloğu parçasında, önceden eğitilmiş bir ağ olan “alexnet” ağı “net” değişkenine atanır. Ardından, “alexnet” ağının son üç katmanı hariç tüm katmanları “layersTransfer” değişkenine atanır. “alexnet” ağı 25 katmandan oluştuğu için “layersTransfer” değişkenine ilk 22 katman atanmış olur. “numel(categories(trainingImages.Labels))” komutu yardımıyla “trainingImages”

değişkeni içerisindeki sınıf sayısı belirlenir ve “numClasses” değişkenine atanır. “alexnet” ağının ilk 22 katmanını temsil eden “layersTransfer” ve “alexnet” ağının son 3 katmanı (layersTransfer, fullyConnectedLayer, softmaxLayer), “layers” değişkenine atanır. “floor(numel(trainingImages.Labels)/miniBatchSize)” komutu yardımıyla eğitim kümesi resimleri sayısı “miniBatchSize” değişkenine bölünerek evre başına düşen işlem adımı sayısı 12 olarak belirlenir ve bu değer “numIterationsPerEpoch” değişkenine atanır. Ardından, eğitim seçenekleri belirlenir ve “options” değişkenine atanır.

```

netTransfer = trainNetwork(trainingImages,layers,options);
predictedLabels = classify(netTransfer,validationImages);
idx = randperm(numel(validationImages.Labels),9);
figure
for i = 1:numel(idx)
    subplot(3,3,i)
    I = readimage(validationImages,idx(i));
    label = predictedLabels(idx(i));
    imshow(I)
    title(char(label))
end
accuracy = mean(predictedLabels == validationImages.Labels)

```

Üstteki kod bloğu parçasında, eğitim kümesi resimleri (trainingImages), ağ katmanları (layers) ve eğitim seçenekleri (options); “trainNetwork” komutuyla işleme sokularak eğitime başlanır. Eğitim sonucu oluşan eğitilmiş ağı, “netTransfer” değişkenine atanır. Eğitilmiş “netTransfer” ağı, “classify” komutu, doğrulama kümesi resimlerinin (validationImages) “arizali_izolator” ve “saglam_izolator” sınıflarından hangisine ait olduğunu tespit ederek sonuçları “predictedLabels” değişkenine atar. Doğrulama kümesi resimlerinden (validationImages) 9 tanesi “randperm” komutu ile rastgele seçilir ve seçilen izolatör resimlerinin numaraları “idx” değişkenine atanır. Doğrulama kümesi resimlerinden (validationImages) rastgele seçilen 9 tane izolatör resmi “subplot” komutuyla “3x3” formatında ekrana yansıtılır. Ekrana yansıtılan bu 9 izolatör resminin “arizali_izolator” ve “saglam_izolator” sınıflarından hangisine ait olduğu, izolatör resimlerinin üzerine “title” komutu ile yazdırılır. Rastgele seçilmiş 104 adet

doğrulama kümesi resimlerinden (validationImages) bilgisayarın tespit ettiği sınıflar (predictedLabels) ile bu doğrulama kümesi resimlerinin gerçekte hangi sınıfa ait oldukları (validationImages.Labels), “==” operatörü ile karşılaştırılır. Karşılaştırılan izolatör resimlerinden yüzde kaçının bilgisayar tarafından doğru tespit edildiği “accuracy” değişkenine atanır.

Yukarıdaki kod bloğu bütün olarak çalıştırıldığında eğitim kümesi resimleri (trainingImages) ile doğrulama kümesi resimleri (validationImages) her defasında “randomized” komutuyla rastgele seçime tabi tutulacağı için yüzdeler doğruluk başarımları yani “accuracy” değeri her defasında farklı bir değer verecektir. Bundan dolayı kod bloğu üç kez çalıştırılmış ve üç sonuç da aşağıdaki tablo 2.30’da gösterilmiştir.

Tablo 2.30. Kod bloğunun üç kez çalıştırılmasıyla ortaya çıkan üç ayrı sonucun tablo halinde gösterimi

	Rastgele seçilmiş 52 adet arızalı izolatörden bilgisayarın doğru tespit ettiği arızalı izolatör sayısı	Rastgele seçilmiş 52 adet sağlam izolatörden bilgisayarın doğru tespit ettiği sağlam izolatör sayısı	Bilgisayarın yüzdeler doğruluk başarımları tespit sonuçları yani “accuracy” değeri
Birinci sonuç	37	38	%72,12
İkinci sonuç	41	28	%66,35
Üçüncü sonuç	30	37	%64,42

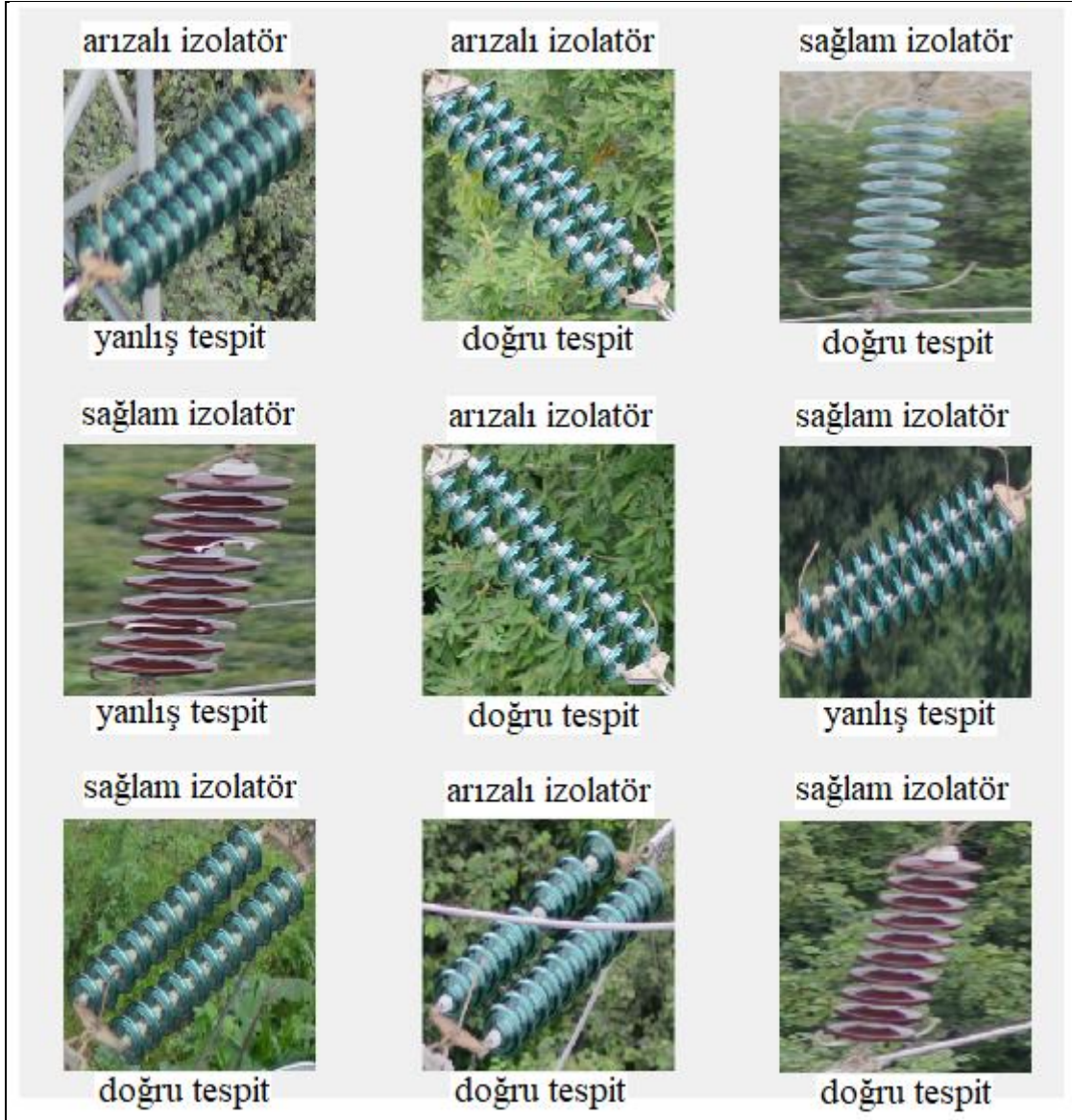
Kod bloğumuzun üç defa çalıştırılması sonucu oluşan üç farklı sonuç için rastgele seçilmiş 52 adetlik doğrulama kümesi resimlerinden (validationImages) rastgele seçilen 9 adedinin bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimleri şekil 2.63, şekil 2.64 ve şekil 2.65’teki gibidir. Her defasında rastgele seçilen bu farklı 9 izolatörün; şekil 2.63’te hepsi, şekil 2.64’te 7 tanesi ve şekil 2.65’te de 6 tanesi doğru tespit edilmiştir. Şekil 2.64’te birinci satırın üçüncü sütunundaki ile üçüncü satırın ikinci sütunundaki izolatörler yanlış tespit edilmiştir. Şekil 2.65’te birinci satırın birinci sütunundaki, ikinci satırın birinci sütunundaki ve ikinci satırın üçüncü sütunundaki izolatörler yanlış tespit edilmiştir.



Şekil 2.63. Yüzdelik doğruluk başarım sonucunun %72,12 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi



Şekil 2.64. Yüzelik doğruluk başarımlarının %66,35 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi



Şekil 2.65. Yüzelik doğruluk başarımlarının %64,42 olduğu durumda, rastgele seçilen 9 adet izolatörün bilgisayar tarafından hangi sınıfa ait olduğunun tespitinin gösterimi.

2.11.5. Öğrenme Transferi (Transfer Learning) ile Eğitilen Ağımız Yardımıyla Etiketleme İşlemine Tabi Tutulmayan Yeni Resimlerdeki İzolatörlerin Arızalı ve Sağlam Olarak Tespiti

Etiketleme işlemi uygulanmış 170 adet resmimizin dışında yeni resimler içerisindeki izolatörlerin arızalı mı yoksa sağlam mı olduğunu tespit etmek için aşağıdaki kod parçası kullanılmıştır [20].

```
load('netTransfer.mat')
validationImages2 = imageDatastore('izolatorler2');
predictedLabels = classify(netTransfer,validationImages2);
```

Yukarıdaki kod parçasında, eğitim aktarımı ile eğitilen “netTransfer.mat” ağıımız “load” komutu ile sisteme yüklenir. Ardından, etiketleme işlemine tabi tutulmayan yeni resimlerimizin içerisindeki izolatörlerden rastgele seçtiğimiz ve “izolatorler2” klasörüne kaydettiğimiz 21 adet izolatör (15 tanesi sağlam ve 6 tanesi arızalı), “imageDatastore” komutu kullanılarak “validationImages2” değişkenine atanır. Sonrasında, eğitilmiş “netTransfer” ağıımız yardımıyla “classify” komutu, bu 21 adet izolatörün, “arizali_izolator” ve “sağlam_izolator” sınıflarından hangisine ait olduğunu tespit ederek sonuçları “predictedLabels” değişkenine atar. Şekil 2.66’da görüldüğü gibi, bilgisayarın tespit ettiği sınıflar (“predictLabels” başlığı altındakiler) ile gerçek sınıfların (“validationImages.Files” başlığı altındakiler), çalışma uzayı çıktılarını yan yana getirilerek karşılaştırma yapılmıştır. Bu karşılaştırma sonucunda 21 adet izolatörden 13 adedinin sınıfının doğru tespit edildiği şekil 2.66’da görülmektedir. Böylece doğruluk başarımlarımız yüzdesi, %61,90 olarak tespit edilmiştir ((13/21)x100=%61,90).

predictedLabels		validationImages.Files	
21x1 categorical		validationImages.Files	
1		1	
1 arizali_izolator	⇒ Doğru ⇐	1 C:\Users\Hp\Desktop\	\DSC_8868_arizali_1.png
2 saglam_izolator	⇒ Doğru ⇐	2 C:\Users\Hp\Desktop\	\DSC_8868_saglam_1.png
3 arizali_izolator	⇒ Yanlış ⇐	3 C:\Users\Hp\Desktop\	\DSC_8868_saglam_2.png
4 arizali_izolator	⇒ Yanlış ⇐	4 C:\Users\Hp\Desktop\	\DSC_8868_saglam_3.png
5 arizali_izolator	⇒ Yanlış ⇐	5 C:\Users\Hp\Desktop\	\DSC_8868_saglam_4.png
6 arizali_izolator	⇒ Yanlış ⇐	6 C:\Users\Hp\Desktop\	\DSC_8868_saglam_5.png
7 saglam_izolator	⇒ Yanlış ⇐	7 C:\Users\Hp\Desktop\	\DSC_8871_arizali_1.png
8 saglam_izolator	⇒ Doğru ⇐	8 C:\Users\Hp\Desktop\	\DSC_8871_saglam_1.png
9 saglam_izolator	⇒ Doğru ⇐	9 C:\Users\Hp\Desktop\	\DSC_8871_saglam_2.png
10 saglam_izolator	⇒ Doğru ⇐	10 C:\Users\Hp\Desktop\	\DSC_8884_saglam_1.png
11 saglam_izolator	⇒ Doğru ⇐	11 C:\Users\Hp\Desktop\	\DSC_8884_saglam_2.png
12 arizali_izolator	⇒ Yanlış ⇐	12 C:\Users\Hp\Desktop\	\DSC_8884_saglam_3.png
13 arizali_izolator	⇒ Doğru ⇐	13 C:\Users\Hp\Desktop\	\DSC_8895_arizali_1.png
14 saglam_izolator	⇒ Doğru ⇐	14 C:\Users\Hp\Desktop\	\DSC_8895_saglam_1.png
15 arizali_izolator	⇒ Doğru ⇐	15 C:\Users\Hp\Desktop\	\DSC_8897_arizali_1.png
16 arizali_izolator	⇒ Doğru ⇐	16 C:\Users\Hp\Desktop\	\DSC_8897_arizali_2.png
17 arizali_izolator	⇒ Yanlış ⇐	17 C:\Users\Hp\Desktop\	\DSC_8980_saglam_1.png
18 saglam_izolator	⇒ Doğru ⇐	18 C:\Users\Hp\Desktop\	\DSC_8980_saglam_2.png
19 arizali_izolator	⇒ Doğru ⇐	19 C:\Users\Hp\Desktop\	\DSC_8984_arizali_1.png
20 saglam_izolator	⇒ Doğru ⇐	20 C:\Users\Hp\Desktop\	\DSC_8984_saglam_1.png
21 arizali_izolator	⇒ Yanlış ⇐	21 C:\Users\Hp\Desktop\	\DSC_8984_saglam_2.png

Şekil 2.66. Etiketleme işlemine tabi tutulmayan yeni resimlerimizin içerisindeki izolatörlerden rastgele seçilen 21 adedi için bilgisayarın arızalı ve sağlam izolatör sınıflarını tespiti

3. SONUÇLAR

On altışar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle ve kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "ImageMetrics" tablolarındaki “GlobalAccuracy” değerleri aynı resimler için karşılaştırıldığında doğruluk başarımlarının genel olarak yükseldiği şekil 3.1’de gösterilmiştir. Bu yükselişin sebebi, ilk sütundaki değerler kırk etiketli resim ile eğitim sonucu oluşurken ikinci sütundaki değerlerin yüz resim ile eğitim sonucu oluşmuş olmasından kaynaklanır. Burada birinci ve ikinci sütundaki sıralama numaraları çalışma uzayındaki “pxdsTest.Files” sıralamasıdır (“pxdsTest.Files” sıralaması, şekil 2.26 ve şekil 2.37’de gösterilmiştir).

On altı resimlik “imdsTest” ve “pxdsTest” resim kümeleri için kırk resimle ve kırkar resimlik “imdsTest” ve “pxdsTest” resim kümeleri için yüz resim ile eğitilmiş ağ kullanılarak oluşturulan tablo 3.1 incelendiğinde kırk resim ile eğitilen ağa göre yüz resim ile eğitilen ağın bize daha yüksek yüzdesel doğruluk başarımları verdiği görülmüştür.

Bilgisayarın izolatör sınıflarını doğru tespit etmedeki başarısını artırmak için sınıf sayısı “arızalı” ve “sağlam” olmak üzere ikiye düşürülmüş ve eğitim aktarımı (transfer learning) yöntemi kullanılarak oluşturulan eğitilmiş ağ yardımıyla izolatörlerin “arızalı” ve “sağlam” sınıflarından hangisine ait oldukları tespit edilmiş. Tablo 3.2’de gösterildiği gibi bu tespit yöntemindeki doğruluk başarımları yüzdeleri %61, %64, %66 ve %72 olarak ölçülmüştür.

	16'şar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 40 resim ile eğitilmiş ağ kullanılarak oluşturulan "ImageMetrics" tablosu		40'ar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 100 resim ile eğitilmiş ağ kullanılarak oluşturulan "ImageMetrics" tablosu	
		GlobalAccuracy		GlobalAccuracy
26'ncı resim	1	0.5056	13	0.5324
19'uncu resim	2	0.5234	16	0.5951
15'inci resim	3	0.5014	18	0.5848
1'inci resim	4	0.6377	20	0.6644
36'ncı resim	5	0.8981	21	0.8711
23'üncü resim	6	0.3988	24	0.7338
2'nci resim	7	0.5964	26	0.6529
4'üncü resim	8	0.7482	27	0.7179
18'inci resim	9	0.8029	28	0.9342
24'üncü resim	10	0.3502	31	0.2738
39'uncu resim	11	0.1099	33	0.0856
13'üncü resim	12	0.9712	34	0.9988
9'uncu resim	13	0.4086	35	0.4850
20'nci resim	14	0.5492	37	0.5933
10'uncu resim	15	0.7578	39	0.8098
12'nci resim	16	0.5842	40	0.5627

Şekil 3.1. On altışar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için kırk resimle; kırkar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "Resim ölçümleri (ImageMetrics)" tablolarındaki "Genel Doğruluk (GlobalAccuracy)" değerlerinin aynı resimler için karşılaştırılması

Tablo 3.1. On altışar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için kırk resimle ve kırkar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için yüz resimle eğitilmiş ağ kullanılarak oluşturulan "Doğruluk" ve "Genel Doğruluk" tablosu

	Doğruluk (Accuracy)		Genel Doğruluk (GlobalAccuracy)	
	16'şar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 40 resim ile eğitilmiş ağ kullanılarak oluşturulan sonuçlar	40'ar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 100 resim ile eğitilmiş ağ kullanılarak oluşturulan sonuçlar	16'şar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 40 resim ile eğitilmiş ağ kullanılarak oluşturulan sonuçlar	40'ar resimlik "imdsTest" ve "pxdsTest" resim kümeleri için 100 resim ile eğitilmiş ağ kullanılarak oluşturulan sonuçlar
arizali_cam	%27,916	%38,807	%53,267	%58,145
arizali_pors	%1,4385	%4,4785		
saglam_cam	%58,373	%63,147		

Tablo 3.1'in devamı

saglam_pors	%73,957	%81,932		
belirsiz	%59,331	%65,977		

Tablo 3.2. Eğitim aktarımı (transfer learning) yöntemi ile elde edilen doğruluk başarımları sonuçları tablosu

		Rastgele seçilmiş 52 adet arızalı izolatörden bilgisayarın doğru tespit ettikleri	Rastgele seçilmiş 52 adet sağlam izolatörden bilgisayarın doğru tespit ettikleri	Bilgisayarın doğruluk başarımları tespit sonuçları
Etiketleme işlemine tabi tutulmuş resimlerdeki izolatörün bilgisayar tarafından arızalı ve sağlam olarak tespit edilmesindeki doğruluk başarımları yüzdesi	Birinci sonuç	37	38	%72,12
	İkinci sonuç	41	28	%66,35
	Üçüncü sonuç	30	37	%64,42
		Rastgele seçilmiş 6 adet arızalı izolatörden bilgisayarın doğru tespit ettiği arızalı izolatör sayısı	Rastgele seçilmiş 15 adet sağlam izolatörden bilgisayarın doğru tespit ettiği arızalı izolatör sayısı	Bilgisayarın yüzdelerle doğruluk başarımları tespit sonucu
Etiketleme işlemine tabi tutulmamış resimlerdeki izolatörün bilgisayar tarafından arızalı ve sağlam olarak tespit edilmesindeki doğruluk başarımları yüzdesi		5	8	%61,90

İletim hatları boyunca çekilen binlerce izolatör fotoğrafının bilgisayar ekranında insan gözü ile tek tek incelenmesiyle izolatör hasarlarının (arızalarının) tespit edilmesi

yerine, derin öğrenme (deep learning) yöntemi kullanılarak bilgisayar üzerinden çok daha kısa sürede ve kolay bir şekilde izolatör hasarlarının sınıflar halinde otomatik tespit edilmesi matlab programı kullanılarak test edilmiş ve ortaya çıkan sonuçlar incelenmiştir. İzolatör sınıflarını tespit etmedeki doğruluk başarımların yüzdeleri; beş sınıf ve kırk resim ile eğitilmiş ağımız için %53, beş sınıf ve yüz resim ile eğitilmiş ağımız için %58, iki sınıf ve 104 resim ile eğitim aktarımı (transfer learning) yöntemi kullanılarak eğitilmiş ağımız için %72 olarak bulunmuştur.

Bilgisayara matlab programı aracılığıyla yükleyeceğimiz resimlerde bilgisayarın izolatör hasarlarına ait sınıfları tespit etmesindeki doğruluk derecesi önce kırk adet etiketlenmiş resim ile eğittiğimiz ağ için daha sonra yüz adet etiketlenmiş resim ile eğittiğimiz ağ için gözlemlenmiştir. Eğiteceğimiz ağımız için ne kadar çok resim etiketlersek ve etiketlediğimiz resimlerdeki çeşitlilik (izolatörlerin duruş açıları, renkleri, şekilleri, uzunlukları, vb.) ne kadar çok ise ağımızın, eğitimden sonra karşısına çıkacak yeni resimlere karşı daha başarılı sonuçlar verdiği sonucuna ulaşılmıştır. İzolatör sınıflarının sayısı azaltılarak ve eğitim aktarımı (transfer learning) yöntemi kullanılarak eğitilen ağ yardımıyla bilgisayarın izolatör sınıflarını doğru tespit etmedeki başarısının daha da yükseldiği sonucuna varılmıştır.

Bu çalışma izolatör arıza sınıflarını tespit etmek için yapılmıştır, bununla birlikte deep learning çok geniş bir alanda kullanılmakta olup istenilen farklı resimler için çeşitli sınıflar belirlenerek çok daha farklı durumların tespit edilmesinde kullanılmak üzere bu çalışma daha da geliştirilebilir.

4. KAYNAKLAR

1. Kikuchi, T., Nakauchi, H., Matsuoka, R. ve Akizuki, M., Remote Sensing System for Faulty Suspension Insulator Units, Proceedings of the 5th International Conference on Properties and Applications of Dielectric Materials, 2 (1997) 766-769.
2. Liang, K., Wang, J., Liu, K. ve Wang, L., Research of Acoustic Vibration Technique in Online Defects Detection in Porcelain Post Insulator, 2015 IEEE International Magnetism Conference (INTERMAG), Mayıs 2015, Beijing, 1-1.
3. Frederick, W., Substation Insulator Failure Prevention by Ultrasonic Corona Detection, IEEE Transactions on Industry Applications, 8 (1972) 82 - 83.
4. Wong, K. L., Application of Very-High-Frequency (VHF) Method to Ceramic Insulators, IEEE Transactions on Dielectrics and Electrical Insulation, 11,6 (2004) 1057-1064.
5. Qaddoumi, N., El-Hag, A. H., Hosani, M. A., Mansouri, I. A. ve Ghufli, H. A., Detecting Defects in Outdoor Non-Ceramic Insulators using Near-Field Microwave Non-Destructive Testing, IEEE Transactions on Dielectrics and Electrical Insulation, 17,2 (2010) 402-407.
6. Yamaguchi, H., Nonaka T., Sato F., Matsuki H. ve Sato T., The Search System for Cracks in the Distribution Line using Magnetic Field, 2005 IEEE International Magnetism Conference (INTERMAG), Nisan 2005, Nagoya, 219-219.
7. Gorur, R. S. ve Sivasubramaniyam, S., Computation of Defect-Induced Electric Fields on Outdoor High Voltage Ceramic and Non-Ceramic Insulators, 2002 Annual Report Conference on Electrical Insulation and Dielectric Phenomena, Ekim 2002, Quintana Roo, 319-322.
8. Cho, B., Byun, S., Park, J., ve Kim, J., Development of Automatic Inspection Robot for Live-Line Insulators, ESMO 2006 - 2006 IEEE 11th International Conference on Transmission & Distribution Construction, Operation and Live-Line Maintenance, Ekim 2006, Albuquerque, 1-5.
9. Tao, X., Zhang, D., Wang, Z., Liu, X., Zhang, H. ve Xu, D., Detection of Power Line Insulator Defects Using Aerial Images Analyzed With Convolutional Neural Networks, IEEE Transactions on Systems, Man and Cybernetics: Systems, 48,10 (2018) 1-13.
10. Shihab, S. ve Wong, K. L., Detection of Faulty Components on Power Lines using Radio Frequency Signatures and Signal Processing Techniques, 2000 IEEE Power Engineering Society Winter Meeting (Conference Proceedings), 4 (2000) 2449-2452.
11. Yang, W., Spatial Scanning-Probe Array System for Silicon-on-Insulator Integrated Circuits, 2008 51st Midwest Symposium on Circuits and Systems (IEEE Conferences), Ağustos 2008, Knoxville, 910-913.
12. Birlasekaran, S. ve Li, H. J., Detection of Faulty Insulators on Power Transmission Line, 2000 IEEE Power Engineering Society Winter Meeting (Conference Proceedings), 4 (2000) 2817-2821.

13. Liu, P., Zhang, Z., Cheng, L., Mei, H. ve Wang, L., Electric Field Distribution Characteristics of Composite Insulator with Internal Defect, 2015 IEEE 11th International Conference on the Properties and Applications of Dielectric Materials (ICPADM), Temmuz 2015, Sydney, 1011-1014.
14. Lasalvia, L. A. B., Florentine, M. T. B., Ferreira, T. V., Germano, A. D., da Costa, E. G., Intelligent Acoustic Detection of Defective Porcelain Station Post Insulators, IEEE Electrical Insulation Conference (EIC), Haziran 2015, Seattle, 118-122.
15. Anjum, S., Jayaram, S., El-Hag, A. ve Jahromi, A. N., Detection and Classification of Defects in Ceramic Insulators using RF Antenna, IEEE Transactions on Dielectrics and Electrical Insulation, 24,1 (2017) 183-190.
16. Nyamupangedengu, C., Luhlanga, L. P. ve Letlape, T., Acoustic and HF Detection of Defects on Porcelain Pin Insulators, 2007 IEEE Power Engineering Society Conference and Exposition in Africa (PowerAfrica), Temmuz 2007, Johannesburg, 1-5.
17. [www.derinogrenme.com/Derin Öğrenme \(Deep Learning\) Nedir? 21 Temmuz 2015](http://www.derinogrenme.com/Derin%20Ogrenme%20(Dep%20Learning)%20Nedir%2021%20Temmuz%202015).
18. Matlab R207b library/Examples/Computer Vision System Toolbox/MATLAB Examples/Object Detection and Recognition/Semantic Segmentation Using Deep Learning.
19. Matlab R207b library/Examples/Neural Network Toolbox/Transfer Learning Using AlexNet.
20. Matlab R207b library/Examples/Computer Vision System Toolbox/MATLAB Examples/Object Detection and Recognition/Image Category Classification Using Deep Learning.

ÖZGEÇMİŞ

Ahmet TURFANDA, 1984 yılında Trabzon'da doğdu. İlkokul, ortaokul ve lise eğitimini Trabzon'da; üniversite eğitimini İstanbul'da tamamladı. Tevfik Serdar Anadolu Lisesi'nden 2002; İstanbul Üniversitesi Elektrik-Elektronik Mühendisliği Bölümünden 2010'da mezun oldu. İyi derecede İngilizce bilmektedir.

