



T.C.

ALANYA ALAADDİN KEKUBAT ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

METALURJİ VE MALZEME MÜHENDİSLİĞİ

ANA BİLİM DALI

**SEM (TARAMALI ELEKTRON MİKROSKOBU)
GÖRÜNTÜLERİNİN YAPAY ZEKÂ METOTLARI İLE
İNCELENMESİ**

Yazarın Adı SOYADI

Ayşe DEMİRKAN

Danışman

Doç. Dr. İsmail TOPCU

ALANYA

2023

Ayşe DEMİRKAN

SEM GÖRÜNTÜLERİNİN YAPAY ZEKÂ METOTLARI İLE İNCELENMESİ

ALKÜ 2023

T.C.
ALANYA ALAADDİN KEYKUBAT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SEM (TARAMALI ELEKTRON MİKROSKOBU) GÖRÜNTÜLERİNİN
YAPAY ZEKÂ METOTLARI İLE İNCELENMESİ

Yüksek Lisans Tezi

Ayşe DEMİRKAN

Anabilim Dalı: Metalurji ve Malzeme Mühendisliği
Program Adı: Nanobilim ve Nanoteknoloji

Danışman

Doç. Dr. İsmail TOPCU

ALANYA

2023

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Alanya Alaaddin Keykubat Üniversitesi tarafından kullanılan “bilimsel intihal tespit programıyla tarandığını ve intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Ayşe DEMİRKAN

TEŞEKKÜR SAYFASI

Günümüz, bilgi ve teknolojinin, özellikle nanoboyutta yer alan incelemelerin giderek her alanda kendini gösterdiği; ayrıca yapay zekânın bu alanda etkilerinin günden güne önem kazandığı bir zaman dilimidir.

Bu kapsamda bitirme tezimin ilk bölümünde nanoteknolojinin ne olduğuna ve gelecekte nanoteknoloji bulgularına yer verildi. Ardından yapay zekâ ve kullanılan yöntemlerde günümüzde en çok tercih edilen sistem ve yöntemlerden bahsedildi. Literatür kısmında, deneysel çalışmada kullanılan malzemelerin özelliklerine kısaca yer verilerek; sonrasında yöntem kısmında kullanılan yapay zekâ teknikleri açıklandı. Bulgular kısmında yapılan deneysel çalışma neden ve sonuçları ile açıklandı. Sonuç ve öneriler kısmında nanoteknoloji ve yapay zekânın malzeme bilimi açısından önemi, deneysel sonuçlarda elde edilen sayısal değerler kıyaslanarak en iyi sonuç veren değer diğer sonuçlarla kıyaslaması yapılarak; gelecek bulguların iyileştirilmeleri kısmında tartışmaya yer verilerek bitirildi.

Yüksek lisans tezimin yazım aşamasında bilgi ve tecrübelerini esirgemeyen çok saygı değer danışman hocam Sayın Doç. Dr. İsmail TOPCU'ya teşekkür ederim.

Tezimin içeriğinde yer alan yapay zekâ oluşumunda bilgi ve kod sistemini oluşturma ve düzenleme sürecinde desteğini esirgemeyen, bilgi ve düşüncelerinden ilham aldığım değerli hocam Sayın Prof. Dr. Ulaş BAĞCI'ya teşekkürlerimi sunarım.

Son olarak, girdiğim bilimsel yol ve süreçte hep yanımda olup; bana ve çalışmalarına saygı duyarak dürüstlükleri ile bugünlere gelmemi sağlayan saygıdeğer babam, annem, kardeşim ve ailesi ile yiğenime sonsuz sevgilerimle...

Ayşe DEMİRKAN

ÖZET

SEM (TARAMALI ELEKTRON MİKROSKOBU) GÖRÜNTÜLERİNİN YAPAY ZEKÂ METOTLARI İLE İNCELENMESİ

Ayşe DEMİRKAN

Metalurji ve Malzeme Mühendisliği Anabilim Dalı

Alanya Alaaddin Keykubat Üniversitesi, Lisansüstü Eğitim Enstitüsü

Ocak,2023 (92 Sayfa)

Günümüzde, Nanoteknoloji ve Nanobilimin, Yapay Zekâ ile çalışma niteliğinin günden güne artması; malzeme biliminin önem kazanmasında etkili olmaktadır.

Tezin SEM Görüntülerinin Yapay Zekâ Metotları ile İncelenmesi, multidisipliner bir alanı ifade etmektedir.

Tezin içeriği, Nanoteknoloji ve Yapay Zekâ alt başlıklarından oluşmaktadır. Deneysel kısımda kullanılan verilerin oluşumu; halka açık erişim olarak sunulan 22.000 SEM verisinin; Trieste'deki CNR-IOM'un TASC laboratuvarında bulunan ZEISS SUPRA 40 çözünürlüklü cihaz ile 5 yıl boyunca 100 bilim insanının çalışmaları sonucunda elde edildiği bilinmekle birlikte; deneysel çalışma için oluşturulan prototipte yer alan verilerin seçimi için; tek tek çözünürlüğü, görüntü boyutu ve kalitesine göre incelenmesinin ardından, dikkate alınan özellik, görüntü kalitesi olmakla birlikte; bu verilerin oluşumunda 100 görüntü verisi elle manuel olarak nano boyutta ve mikro boyutta seçilip, düzenlendikten sonra; toplamda 1000 görüntü verisi 10 veri seti içerisinde oluşturuldu. Devamında makine öğrenmesi yönteminde yer alan derin öğrenme tekniklerinden CNN sınıflandırma tekniği kullanılarak yapay zekâ eğitimi gerçekleştirildi.

Deneysel çalışma için python kütüphanesinden yararlanıldı. Ardından hazırlanan verilerin "RELU-softmax" aktivasyon fonksiyonları ile; Epoch değerleri RMSprop ile 25 ve 100 olarak ayarlanarak eğitim tamamlandı. Bu eğitimde RMSprop yerine, Adam = 0.000001 fonksiyonu kullanılarak tekrar oluşturulan kodlarla, öğrenme hızındaki artış-azalış değerlerini gözlemlemek adına eğitim tekrarlandı. Veri sonuçlarının eğitim parametrelerine uygunluk düzeyine göre sınıflandırma açısından Model doğruluk oranlarının tablolandırıldı. En iyi sonuç

deęerinin yorumlaması yapıldı. Burada kullanılan yaklaşım, çok çeşitli nanobilim kullanım durumlarına uygulanabilen, nanomalzemelerin belirli özelliklerini çözmek için uygun parametreler geliştirilecek şekilde ayarlanarak yeni yöntem ve araçların uygulanmasına giden yolu makine öğrenmesinin ürüne dönüşüm boyutunda; malzeme bilimi kütüphanesi oluşturmak amacıyla kullanımının, yapay zekâ ve nanoteknoloji alanındaki gelecek çalışmalarının önünü açabilmesine katkısı olabilir.

Anahtar Kelimeler: Nanoteknoloji, Nanobilim, Yapay Zekâ, SEM, Malzeme Bilimi

ABSTRACT

EXAMINATION OF SEM (SCANNING ELECTRON MICROSCOPE) IMAGES BY ARTIFICIAL INTELLIGENCE METHODS

Ayşe DEMİRKAN

Department of Metallurgical and Materials Engineering

Graduate School of Alanya Alaaddin Keykubat University,

January, 2023 (92 Pages)

Today, the quality of Nanotechnology and Nanoscience to work with Artificial Intelligence is increasing day by day; It is effective in gaining importance of materials science.

Examination of SEM Images of the thesis with Artificial Intelligence Methods represents a multidisciplinary field.

The content of the thesis, It consists of sub-titles of Nanotechnology and Artificial Intelligence. The formation of the data used in the experimental part; 22,000 SEM data available as public access; Although it is known that it was obtained as a result of the work of 100 scientists for 5 years with the ZEISS SUPRA 40 resolution device in the TASC laboratory of CNR-IOM in Trieste; for the selection of data contained in the prototype created for experimental work; After examining the resolution, image size and quality one by one, the feature taken into consideration is image quality; In the formation of these data, after 100 image data are manually selected and arranged in nano and micro dimensions; A total of 1000 image data was created in 10 data sets. Afterwards, artificial intelligence training was carried out using the CNN classification technique, one of the deep learning techniques in the machine learning method.

The python library was used for the experimental study. Then, with the “RELU-softmax” activation functions of the prepared data; the training was completed by setting the epoch values to 25 and 100 with RMSprop. In this training, the training was repeated to observe the increase- decrease values in the learning speed with the renewed codes using the Adam=0.000001 function instead of RMSprop. Model accuracy rates are tabulated in terms of classification according to the level of fit of data results with training parameters. The interpretation of the best result value was made.

The approach used here is in the machine learning-to-product dimension, which leads to the application of new methods and tools by adjusting to develop appropriate parameters to solve specific properties of nanomaterials, which can be applied to a wide variety of nanoscience use cases; Its use to create a materials science library may contribute to paving the way for future studies in the field of artificial intelligence and nanotechnology.

Keywords: Nanotechnology, Nanoscience, Artificial Intelligence, SEM, Materials Science.

İÇİNDEKİLER

İÇ KAPAK SAYFASI	
ONAY SAYFASI.....	i
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	ii
TEŞEKKÜR SAYFASI.....	iii
ÖZET.....	iv
ABSTRACT.....	vi
İÇİNDEKİLER SAYFASI.....	viii
TABLOLAR LİSTESİ.....	x
ŞEKİLLER LİSTESİ.....	xi
KISALTMALAR.....	xiv
1. GİRİŞ.....	1
1.1. Nanoteknoloji Genel Bakış.....	2
1.1.1. Nanoteknoloji Nedir?.....	3
1.1.2. Nanoteknolojinin Önemi.....	5
1.1.3. Günümüzde Nanoteknolojinin Kullanım Alanları Nelerdir?.....	6
1.1.4. SEM(Scanning Electron Microscopy) ve Kullanıldığı Yerler.....	8
1.2. AI (Artificial Intelligence).....	10
1.2.1. AI Nedir?.....	10
1.2.2. AI'nın Amacı.....	16
1.2.3. ML (Machine Learning) Nedir?.....	17
1.2.4. DL (Deep Learning) Nedir?.....	18
2. LİTERATÜR.....	21
2.1. CNN (Convolutional Neural Network).....	21
2.1.1. Tensorflow.....	23
2.1.2. Keras.....	23
2.1.3. Matplotlib.....	23
2.1.4. Train Veri Kümesi.....	23
2.1.5. Validation Veri Kümesi.....	23
2.1.6. Test Veri Kümesi.....	23
2.2. Deneysel Olarak Kullanılan 10 Veri Setinin Nanoteknolojideki Yeri ve Önemi.....	23
2.2.1. Lifler (Fibres).....	23
2.2.2. Nanofiberler (Nanowires).....	25

2.2.3. Desen Yüzeyi (Patterned_surface).....	26
2.2.4.Film Kaplı Yüzey (Films_Coated_Surface).....	27
2.2.5.Biyolojik (Biological).....	28
2.2.6.Gözenekli Sünger (Porous_Sponge).....	29
2.2.7. Toz (Powder).....	30
2.2.8.Uçlar (Tips).....	31
2.2.9.Partikül (Particles).....	32
2.2.10.Mems Cihazları ve Elektrotları (MEMS_devices_and_electrodes).....	34
3. YÖNTEM.....	36
4. BULGULAR.....	38
5. TARTIŞMA, SONUÇ VE ÖNERİLER.....	46
6. KAYNAKÇA.....	54
7. EKLER.....	63
EK 1.....	63
EK 2.....	68
EK 3.....	79
EK 4.....	85
ÖZGEÇMİŞ.....	91

TABLÖLAR LİSTESİ

Tablo 1 Deneysel Sonular	44
Tablo 2 Deneysel Sonular	52

ŞEKİLLER LİSTESİ

Şekil 1.1 Altın Nanopartükülleri ve Uygulama Alanları	3
Şekil 1.2 Nanoölçek Gösterimi	4
Şekil 1.3 Yukarıdan aşağı indirgemeci ve Aşağıdan yukarı eklemeli yaklaşım Bilim-Mühendislik Alanları...5	
Şekil 1.4 Karbon nanotüp... ..	6
Şekil 1.5 Nanoteknoloji Uygulama Alanları.....	7
Şekil 1.6 Taramalı Elektron Mikroskobu	9
Şekil 1.7 Taramalı Elektron Mikroskobu ile Uygulama Örneği Örneği.....	10
Şekil 1.8 AI Teknolojisi... ..	11
Şekil 1.9 Yapay Zekâ Geliştirme.....	12
Şekil 1.10 Yapay Sinir Ağı Sınıflandırması.....	13
Şekil 1.11 Tipik Bir Sinir Ağı Mimarisi	14
Şekil 1.12 İki Katmanlı İleri Beslemeli Ağ Mimarisi	15
Şekil 1.13 Bir sinir hücresinin matematiksel gösterimi	15
Şekil 1.14 Yapay Zekâ Uygulamaları.....	16
Şekil 1.15 ML içeriğinde yer alan Yapay Sinir Ağı ve CNN'in Şematik Gösterimi.....	17
Şekil 1.16 Makine öğrenimi teknikleri: Denetimli Öğrenme, Denetimsiz Öğrenme	18
Şekil 1.17 ML tekniklerinden DL ile yüz tanıma örneği.....	20

Şekil 2.1 CNN ağı Görseli.....	22
Şekil 2.2 Lif SEM Görüntü Örneği.....	25
Şekil 2.3 Nanofiber SEM Görüntü Örneği.....	26
Şekil 2.4 Desen Yüzeyi Lif SEM Görüntü Örneği.....	27
Şekil 2.5 Film Kaplı Yüzey SEM Görüntü Örneği.....	28
Şekil 2.6 Biyolojik SEM Görüntü Örneği.....	29
Şekil 2.7 Gözenekli Sünger SEM Görüntü Örneği.....	30
Şekil 2.8 Toz SEM Görüntü Örneği.....	31
Şekil 2.9 Uçlar SEM Görüntü Örneği.....	32
Şekil 2.10 Partikül SEM Görüntü Örneği.....	34
Şekil 2.11 Mems Cihazları ve Elektrotları SEM Görüntü Örneği.....	35
Şekil 4.1 AI Tekniği ile Yapılan Sınıflandırma Toz SEM Görüntü Çıktısı.....	41
Şekil 4.2 AI Tekniği ile Yapılan Sınıflandırmanın Suffle sonucu oluşan Film Kaplı Yüzey SEM Görüntü Çıktısı.....	42
Şekil 4.3 AI tekniği ile Yapılan Sınıflandırmada Gray Skala ile oluşan Lif SEM Görüntü Çıktısı.....	43
Şekil 5.1 AI Tekniği ile Yapılan Sınıflandırma SEM(Toz) Görüntü Çıktısı.....	46
Şekil 5.2 AI Tekniği ile Yapılan Sınıflandırmanın Suffle sonucu oluşan Film Kaplı Yüzey SEM Görüntü Çıktısı.....	47
Şekil 5.3 AI Tekniği ile Yapılan Sınıflandırmada Gray Skala ile oluşan Lif SEM Görüntü Çıktısı.....	47
Şekil 5.4 Deneysel değerlendirmede RMSprop fonksiyonuyla Epoch değerinin 25 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri.....	49
Şekil 5.5 Deneysel değerlendirmede RMSprop fonksiyonuyla Epoch değerinin 100 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri.....	50

Şekil 5.6 Deneysel değerlendirilmede Adam = 0.000001 fonksiyonuyla Epoch değerinin 25 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranı grafik değeri	51
Şekil 5.7 Deneysel değerlendirilmede Adam = 0.000001 fonksiyonuyla Epoch değerinin 100 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranı grafik değeri	52
Şekil Ek 1.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi	65
Şekil Ek 1.2: AI tekniği ile yapılan sınıflandırmada oluşan Lif SEM görüntü çıktısı.....	67
Şekil Ek 2.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi	76
Şekil Ek 2.2: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi.....	78
Şekil Ek 3.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi	82
Şekil Ek 3.2: AI tekniği ile yapılan sınıflandırmada oluşan Lif SEM görüntü çıktısı.....	83
Şekil Ek 4.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistem	87
Şekil Ek 4.2: AI tekniği ile yapılan sınıflandırmada oluşan Lif SEM görüntü çıktısı.....	89

KISALTMALAR

SEM	Taramalı Elektron Mikroskobu
AI	Yapay Zekâ
CNN	Evrişimsel Sinir Ağları
GPU	Grafik İşleme Birimi
YSA	Yapay sinir ağı
FFNN	İleri beslemeli sinir ağı
FITNET	İşlev uydurma sinir ağı
CFNN	Kademeli ileri sinir ağı
GRNN	Genelleştirilmiş regresyon sinir ağı
ML	Makine öğrenmesi
DL	Derin öğrenme
EDS	Elektron Dağılım Spektroskopisi
1D	Tek boyutlu
2D	İki boyutlu
3D	Üç boyutlu
RF	Radyo frekansı
os	Operating System
cv2	Opencv

1.GİRİŞ

Günümüzde, Nanoteknoloji ve Nanobilimin giderek önem kazanması; bu kavramların AI (Artificial Intelligence) ile çalışma niteliğinin farkındalığını günden güne arttırmakla birlikte; özellikle malzeme bilimi mühendisliği alanında ve üretim boyutunda, eşzamanlı kullanılmasıyla, hızlı gelişimlerin önünü açabilir [1-4].

Tezin oluşumunu sağlayan SEM (Scanning Electron Microscopy) görüntülerinin yapay zekâ metotları ile incelenmesi içerik olarak multidisipliner bir alanın yansımasıdır [5-9].

Çalışma genel olarak, literatür de nanoteknolojiden elde edilen bilgilerin ve yapay zekâ çalışmalarının; temel malzeme biliminden, nanotipa, nanorobotik uygulamalara karakterize edilmesi ve raporlanması gereken temel özellikleri ve deneysel koşulların nanobilimsel mimarilerinin derinden ilişkilendirilmesi hakkında rehberlik sağlayabileceği düşünülerek oluşturulmaktadır [1, 3, 5, 10-13].

Tezin içeriği, Nanoteknoloji ve AI alt başlıklarından oluşmaktadır. Burada özellikle iki alanda yapılan çalışmalara kısaca yer verilmesinin gelecek çalışmaların önünün açılması kapsamında değerlendirildiğinde önemli olduğu söylenebilir [1-25].

Nanoteknoloji içerik olarak; Nanoteknoloji nedir? Nanoteknolojinin önemi, Günümüzde nanoteknolojinin kullanım alanları nelerdir? kısımlarında güncel bilgilere yer verilerek; SEM görüntülerinin kullanıldığı yerler, SEM görüntü verilerinin, Nanoteknoloji ve Nanobilim’de uygulama alanlarının belirtilmesiyle yeni ve keşfe değer bir alan olması açısından vurgulanmaktadır [1-25].

Ardından AI içeriği ile AI nedir? kısmında; AI ve uygulama alanlarına kısaca yer verilerek; AI’nın amacı ve alt dallarında kullanılan metotlara değinilmektedir. ML(Machine Learning), DL (Deep Learning) ve literatürde belirtilen CNN (Convolutional Neural Network) metodu ile yapılan sınıflandırma eğitiminde kullanılan; halka açık erişim olarak sunulan 22.000 SEM verisinin; Trieste’deki CNR-IOM’un TASC laboratuvarında bulunan ZEISS SUPRA 40 çözünürlüklü cihaz ile 100 bilim insanı tarafından 5 yıl boyunca elde edildiği bilinmekle birlikte; yapılan deneysel çalışma için prototipi oluşturulan verilerin seçiminde; verilerin tek tek çözünürlüğü, görüntü boyutu ve kalitesine göre incelendikten sonra, dikkate alınan ana özellik, görüntü kalitesi olmakla birlikte; bu verilerden 100’er görüntü verisi elle manuel olarak seçilip; her bir sınıf için tekrar yeni 10 veri seti oluşturularak düzenlendi. Deneysel çalışmada kullanılan SEM veri setleri: Lifler (Fibres); Nanofiberler (Nanowires); Desen

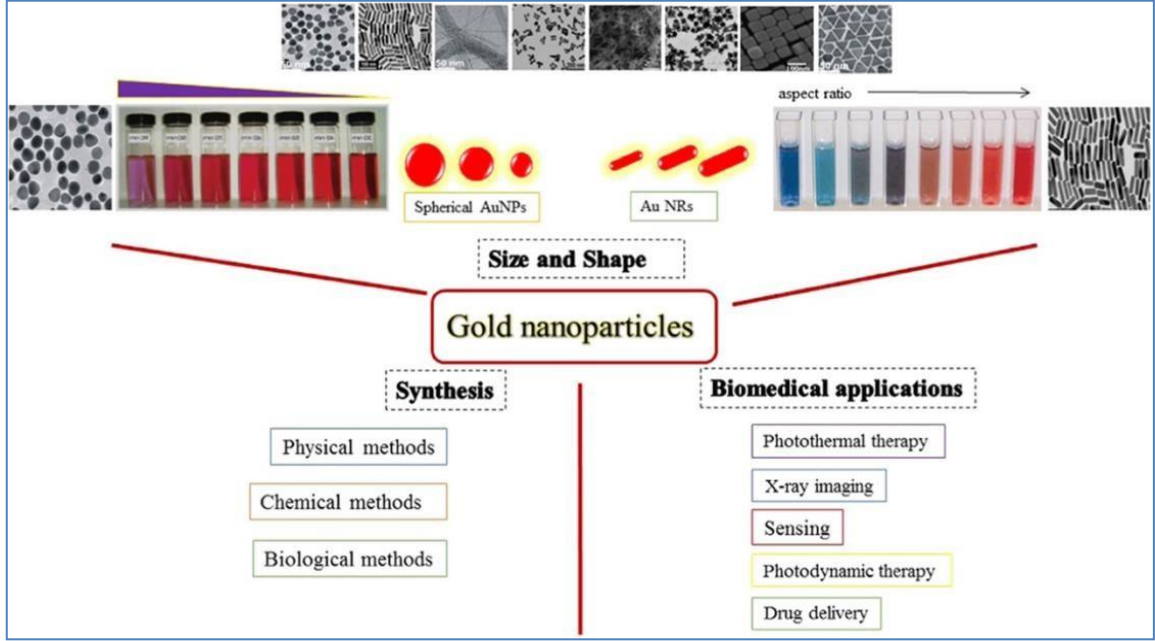
Yüzeyi (Patterned_surface); Film Kaplı Yüzey (Films_Coated_Surface); Biyolojik (Biological); Gözenekli Sünger (Porous_Sponge); Toz (Powder); Uçlar (Tips); Partikül (Particles); Mems Cihazları ve Elektrotları (MEMS_devices_and_electrodes) malzemelerinden oluşmaktadır [25-51, 52-56, 57-70, 71-96, 98-107].

Çalışma, AI metotlarından derin öğrenmenin görüntü sınıflandırmada kullanılan CNN metodu ile gerçekleştirildi. Bunun için Google Colab arayüzü kullanılarak; çalışma zamanı GPU (Graphics Processing Unit) olarak ayarlanarak, python ile deneysel çalışma için gereken kod kısımları oluşturuldu. Ardından, ‘RELU-softmax’ aktivasyon fonksiyonlarıyla veriler; RMSprop ile Epoch değerleri 25 ve 100 olarak ayarlanarak, AI/ML’nin alt kategorisi olan DL’nin görüntü sınıflandırma tekniğinde kullanılan CNN’de eğitimin tamamlanması sağlandı. Bu eğitim, RMSprop yerine Adam = 0.000001 fonksiyonu kullanılarak, öğrenme hızındaki artış-azalış değerlerini gözlemlemek adına tekrarlandı. Sonuç değerlerinin veri sonuçlarıyla eğitim parametrelerine uygunluk düzeyine göre sınıflandırmak, model doğruluk oranlarını tablolaştırılıp; en iyi sonuç değerinin karşılaştırmalı gözlemlenmesi açısından niceliksel boyutta önemlidir. Amaç, SEM verilerinden oluşturulan 10 ayrı anlamsal kategoriyi, AI tekniklerini kullanarak sınıflandırma yaparak, model doğruluk oranlarını gözlemlemek; ayrıca en iyi sonuç değerlerini veren fonksiyonu test etmektir. Kullanılan yaklaşım, çok çeşitli nanobilim kullanım durumlarına uygulanabilir. Nanomalzemelerin belirli özelliklerini çözmek için uygun parametreler geliştirilecek şekilde ayarlanarak yeni yöntem ve araçların uygulanmasına giden yolu açabilir. Malzeme bilimi kütüphanesini oluşturmak için, AI ve nanoteknoloji alanında ürünsel boyutta oluşabilecek gelecek çalışmalara katkı sunabilir [51,52, 55, 56, 95, 96, 98-107].

1.1. Nanoteknolojiye Genel Bakış

Günümüz kapsamında nano boyutun detaylı incelemeleri değerlendirildiğinde Nanoteknolojinin nerede var olmasıdır. Bu, günümüzün nano araştırmacıları için başlıca araştırma boyutlarının bir göstergesidir. İlginç olan durum, modern çağda klasik fizik yasalarının mikro ve nano parçacıklar üzerinde geçerli olmadığı gerçeğidir. Aynı zamanda malzemelerin renk, iletkenlik, manyetizma ve erime noktası gibi bazı temel niteliklerine göre istisnai davranışlar sergilemekle beraber, nanoparçacıklara uyarlandığı anda sabit görünmediğinin bilinmesidir. Bütün olarak bakıldığında bu problemler, nanoteknoloji hakkında yeni bir fikrin ortaya çıkmasına sebep olmakta; öğrenme çalışmalarını makine öğrenmesine dönüştürmek için bazı paradigmanın ve

yaklaşımların kullanılmasıyla çalışmaların titizlikle incelenmesi, sonuçlara yaklaşık olarak çok hızlı bir şekilde ulaşılmasına katkı sunması açısından önemlidir [1-13].

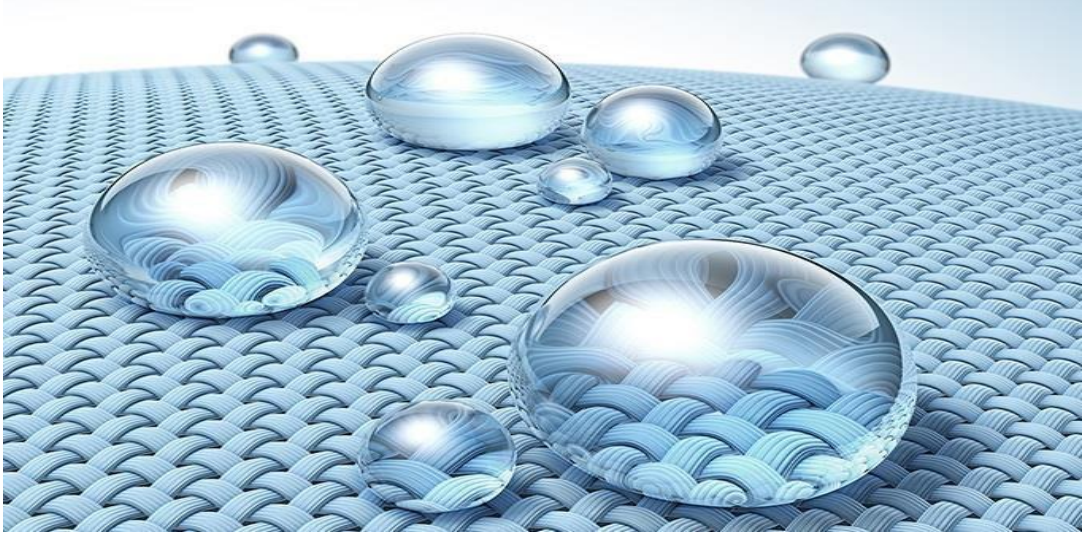


Şekil 1.1 Altın Nanopartükülleri ve Uygulama Alanları [82]

Nanoyapıları karakterizasyon ve boyutlarına göre sınıflandırmanın başka bir yolu, görsel olarak farklı olan sinir ağı tarafından detaylandırılabilmesinin kolay olmasıdır. Bununla birlikte nanoyapıların birden çok işlevselliği şeklinde ifade edilebilmesidir [14,15].

1.1.1. Nanoteknoloji nedir?

Richard Feynman 1959 yılında Caltech'te yaptığı konuşmasında: Nano ölçek boyutlarında olan parçacıkların, maddenin atomik katmanda işlenebilmesiyle beraber; en son teknolojiye sahip nanosensörlerin, gelişmiş çeşitleri içerisinde barındıran malzemelere (kuantum noktaları, nanokabuklar, nanogözenekler, karbon nanotüpler, vb.) dayandığını ifade etmiştir [2,16].

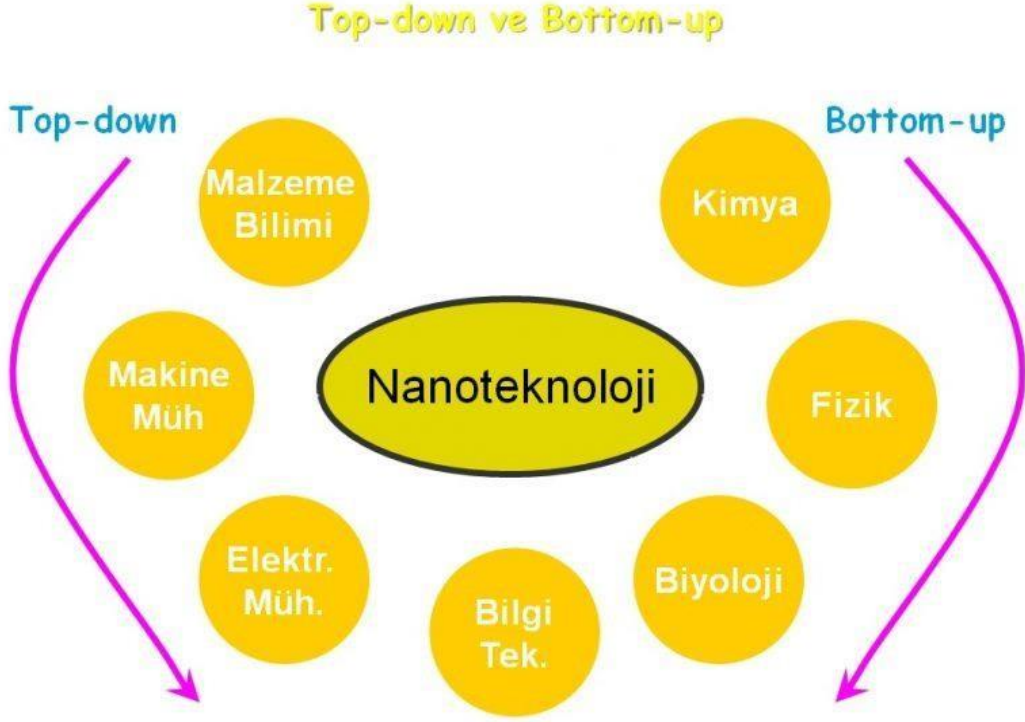


Şekil 1.2 Nanoölçek Gösterimi [83]

Nanoteknoloji, işlevsel en küçük organizasyona sahip olan, en az bir boyutta, nanometre ölçeği olarak metrenin milyarda biridir. Nanobilim, malzemenin cihazlarının tasarımının, sentezinin, karakterizasyonu ile beraber uygulandığı bilim olarak ifade edilebilir [1, 2-3].

Nanoteknoloji mühendisliği, maddenin atomik boyutlu, moleküler düzeyde ve supramoleküler ölçekler ile ölçümünü sağlayan; aynı zamanda hassas bir şekilde kontrolü gerçekleştirilebilen ve moleküler ölçekte fonksiyonel özelliğe sahip olan sistemlerin mühendisliğidir [4].

Kapsam içerisinde; nanoteknoloji ve nanobilim, nano ölçeğin bilimi şeklinde açıklanabilir. Bir nanometre civarındaki nesnelere uygulama alanı bulan bilim dalıdır. Nanometre hassasiyetinde büyük, karmaşık yapılar inşa etme kapasitesinin hızla değişiklik göstermesi; yukarıdan aşağıya indirgeyici yaklaşımlardan ve aşağıdan yukarıya eklemeli yaklaşımlardan oluşmasının yanı sıra, bilim ve mühendislik alanlarında yer almasının önemli bir göstergesidir [1-4].

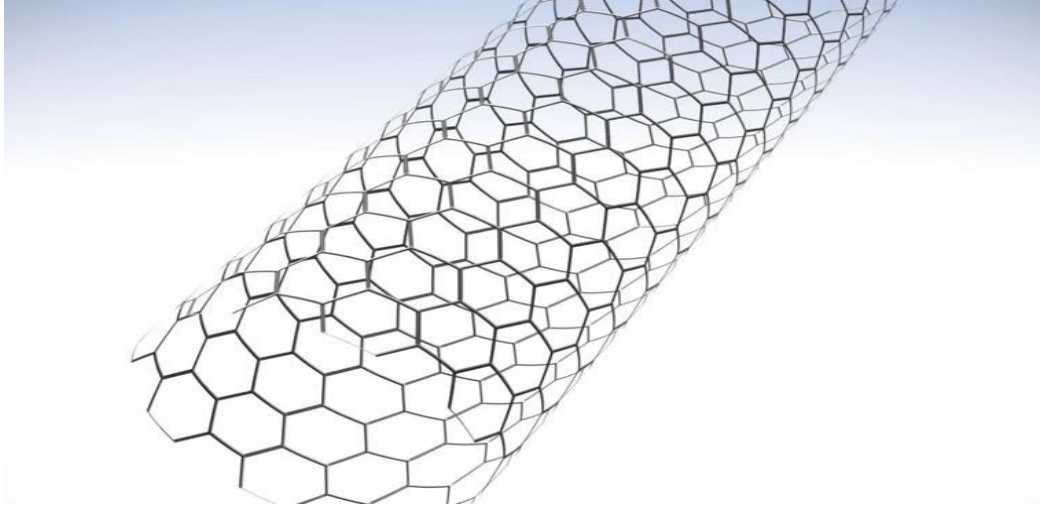


Şekil 1.3 Yukarıdan aşağı indirgemeci ve Aşağıdan yukarı eklemeli yaklaşım Bilim-Mühendislik Alanları [85].

1.1.2. Nanoteknolojinin önemi

Nanoteknolojinin önemi, özellikle uygulama alanının genişliği kapsamında değerlendirildiğinde; tekstil ve plastiklerdeki nanofaz seramik malzemelerden, doğrusal olmayan optik ve optoelektronik alanındaki uygulamalara kadar genişleyen şekilde devam etmektedir. Elektroniklerle beraber, nanoteknolojinin biyoloji alanıyla ilişkilendirilmesinin son derece faydalı olacağı bilinmektedir [5-8].

Nanoteknoloji alanında önemli malzemelerden biri olan Karbon nanotüplerin, özellikle taramalı mikroskoplarda uç olarak ve görüntüleme cihazlarında verimli alan yayıcılar olarak kullanıldığı bilinmektedir [8-16, 17-21].



Şekil 1.4 Karbon nanotüp [87]

Ayrıca Nanoteknolojinin, litografik desenlemeyi kapsayan DNA mikroçip dizilerini geliştirmekle de bağlantısı bulunmaktadır. Bu kapsamda yapılan çalışmalardan özellikle son zamanlarda, benzersiz bir setin tanımlanması için yeni PCR güdümlü yöntem olan bir faj peptit gösterim kitaplığından gümüş ve kobalt nanoparçacıklarına bağlanan dizilerin sayısının geliştirilmektedir. Benzer şekilde, nanoparçacıkların ve nanokapsüllerin kullanımıyla ilaç ve gen aktarımının daha etkili olacağı çalışmalar mevcuttur. Farklı uygulamalar arasında, çevreyi ve canlı sistemleri izlemeye; insan vücudunun parçalarını onarmak veya değiştirmek amacıyla kullanılan protezlerin, Ti-6Al-4V kompozitleri birçok uygulamada kullanılan diğer malzemelere göre daha ucuz ve daha hafif olduğu bilinmektedir. Üretimsel gelişmelerle beraber nanosensörlerin kullanımda yer almaktadır. Bundan ötürü, fizik, kimya, biyoloji, malzeme bilimi ve mühendisliğini de kapsayan gerçekten disiplinler arası bir alan olarak önemlidir [2, 5-16].

1.1.2. Günümüzde nanoteknolojinin kullanım alanları nelerdir?

Son zamanlarda nanoteknoloji mevcut bilim ve teknoloji içerisinde önemli bir kavram olarak yerini almaya başladı [1, 10-13].

Nanoteknolojinin önemli alanlarından biri olan “nano-tıp”; özellikle tıbbi uygulamalarda büyük ilgi görmektedir. Tıp alanında yapılan çalışmalarda, doku ve implant mühendisliği; nanotüp alanında, doku replasmanı, organ rekonstrüksiyonu, moleküler düzeyde teşhis ve tedavi için nanomalzemelerin geliştirildiği bilinmektedir [1, 9-12].

Robotların nano ölçekte incelenmesi nanorobotik birleşik teknolojiye

nanoteknoloji olarak bilinip; nanorobotların nano ölçekte algılaması, harekete geçmesi, sinyal vermesi, bilgi işlemesi, zekâ veya sürü davranışı sergileme yeteneğine sahip olan cihazlarla mümkün olup; insan sağlığını koruma ve iyileştirmeye yönelik deneysel çalışmalar yapılmaktadır [10-12].

Nanotıp, hastalıkları ve travmatik yaralanmaları teşhis etme, tedavi etme, önleme, ağrıyı hafifletmenin yanı sıra, nanoteknoloji ve nanorobotları kullanarak insan sağlığını koruyup iyileştirme yöntemi şeklinde açıklanabilmektedir. Bugün henüz yeteri kadar hızlı, çok sayıda örneği aynı anda değerlendiren, çok düşük konsantrasyonlarda ölçüm olanağı veren cihazlar mevcut değildir. Bunlara ek olarak, biyoçip teknolojisi bu yönde geliştirilen en önemli teknoloji olarak gözüke de henüz istenilen ölçümleri tanımlanan hızda, kalitatif/kantitatif şekilde yapan ve ekonomik (yaygın kullanıma imkân verecek şekilde) olarak çözmekten çok uzak olduğu düşünülmektedir. Ancak gelecekte mutlaka moleküler düzeyde ölçüm yapan, nanoteknolojinin şimdilik çoğu bilinmeyen veya ürüne dönüştürülemeyen avantajlarını kullanan yeni yaklaşımlara gereksinim olması muhtemeldir [1, 10-12].

Bunlara ek olarak, sağlık alanında kullanılabilecek hızlı, kapasitesi yüksek, hassas protein ve DNA tanı sistemlerinin nanoteknoloji kullanılarak geliştirilmesi üzerine çalışmalar devam etmektedir [3, 6, 9].

Son olarak, hedefe yönelik yeni ilaç etken maddelerinin tanımlanması için hızlı nanoteknolojik tarama yöntemlerinin geliştirilmesi önemli bir adım olarak bilgisayarlarda kullanılan yazılım sistemlerinin geliştirilmesi; hedef kitlenin sağlıklı gelişimi için önemli bir adımdır [1-10, 11-40].



Şekil 1.5 Nanoteknolojinin Uygulama Alanları [86]

1.1.4. SEM (Scanning Electron Microscopy) ve kullanıldığı yerler

SEM, malzemelerin yüzey alanlarında oluşan olayları incelemek için sıklıkla kullanılan çok yönlü gelişmiş bir tür elektron mikroskobudur.

Aslında SEM, bir parçacığın yüksek kalitede ve uzaysal çözünürlükte ayrıntılı bir görsel görüntüsünü elde edebilmekle beraber; morfolojisi şekil ve boyutu belirtirken, topografisi bir nesnenin yüzey özelliklerini veya "nasıl görüldüğünü", dokusunu, düzgünlüğünü veya pürüzlülüğünü ölçüm yapan hassas bir araçtır. Bunlara ilave olarak SEM kompozisyonu, malzemeyi oluşturan elementler ve bileşikler anlamına gelip; kristalografisi ise malzemelerdeki atomların düzenlenmesini sağlamakla görevlidir.

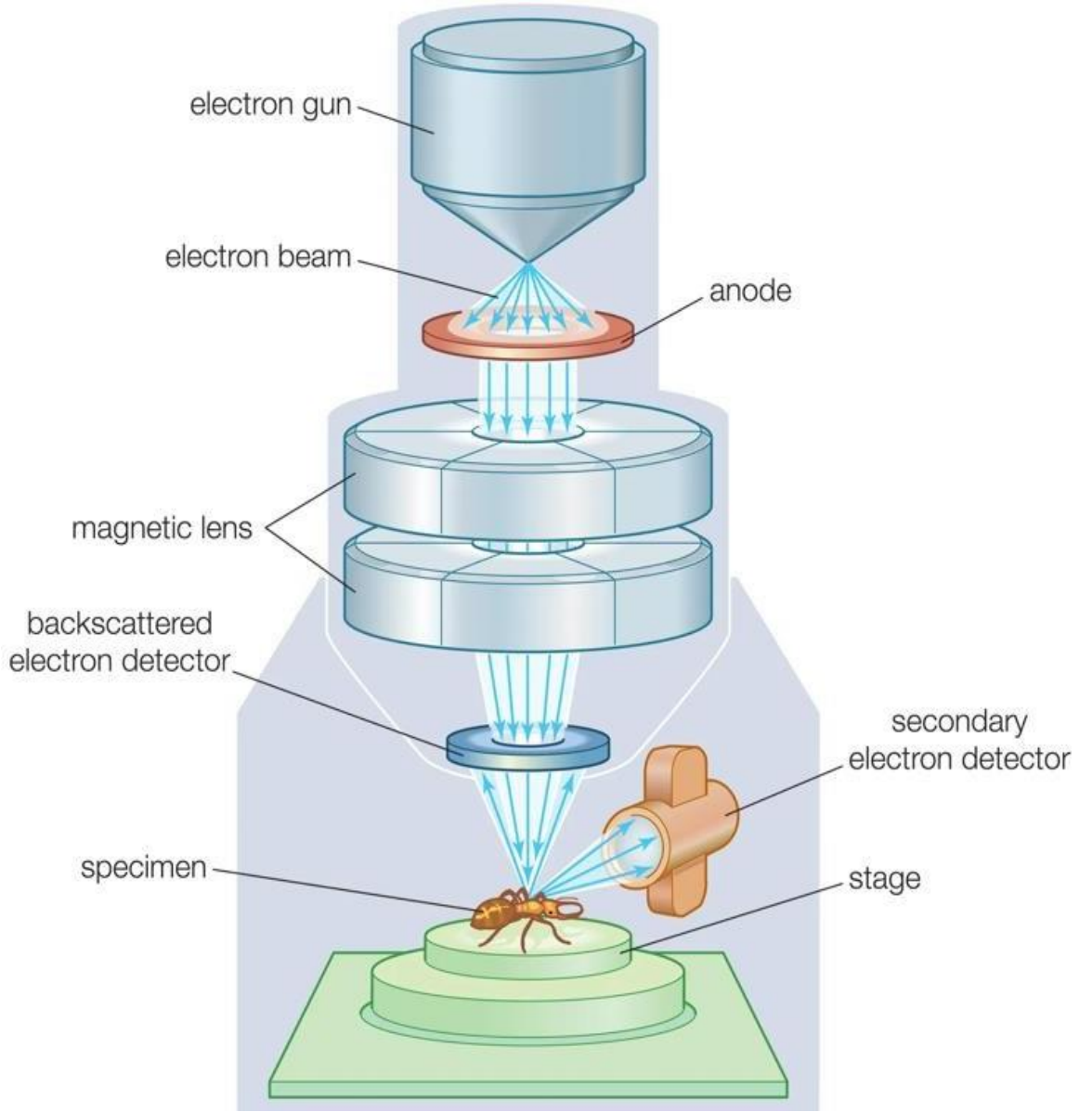
SEM uygulamalarına kısaca değinilecek olursa; SEM, aşağıdaki parametreleri ölçebilir ve analiz edebilir:

- (i) Filmlerin ve ince kaplamaların kalınlığı,
 - (ii) Yüzey morfolojisi ve görünümü,
 - (iii) Boyut ve boyut dağılımı,
 - (iv) Kompozitlerde ve karışımlarda partiküllerin, liflerin,
 - (v) Nanomalzemelerin veya diğer katkı maddelerinin şekli, dağılımı,
 - (vi) Nanometre boyutlu malzemelerin yükseklik ve yan boyutları,
 - (vii) Köpük malzemelerde hücre boyutu ve boyut dağılımı,
 - (viii) Nano ve mikro malzemelerin kimyasal bileşimi ve element analizi,
- kırılma ve yapısal kusur analizi şeklinde geniş bir alana sahiptir.



Şekil 1.6 Taramalı Elektron Mikroskobu [102]

Yukarıdaki kısımlara ek olarak, nanomalzemelerin morfolojisi, düzenlenmesi ve bileşimi hakkında kapsamlı bilgi veren çok amaçlı, tahribatsız bir araçtır ifadesi SEM'i en iyi şekilde özetleyebilir [20-24].

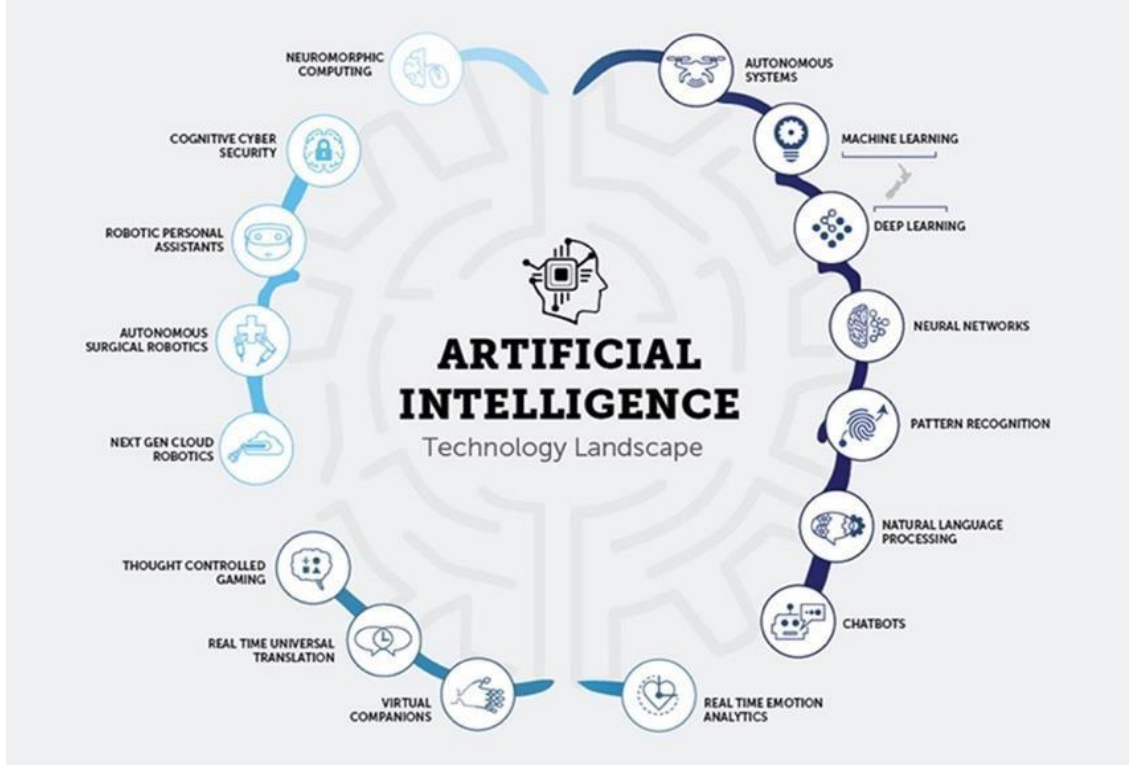


Şekil 1.7 Taramalı Elektron Mikroskobu ile Uygulama Örneği [88]

1.2.AI(Artificial Intelligence)

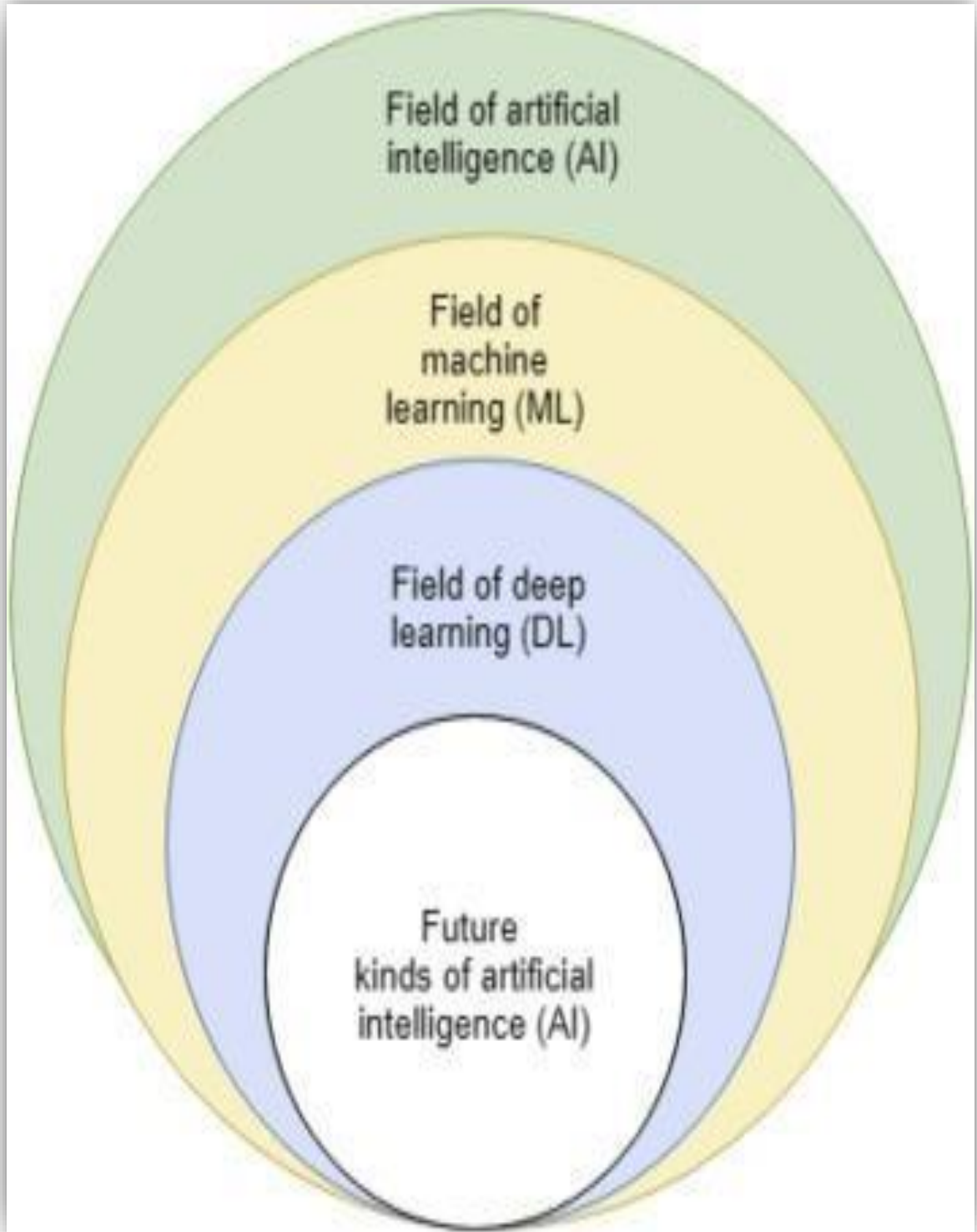
1.2.1. AI(Artificial Intelligence) nedir?

AI, teoride görevleri yerine getirip; insan zekasını taklit eden ve topladıkları bilgilerden yinelemeli olarak kendilerini iyileştirebilen makineler anlamına gelir. Özellikle akıllı bilgisayar programları yapabilecek insan gücünün bir araya gelmesiyle oluşur [25-57].



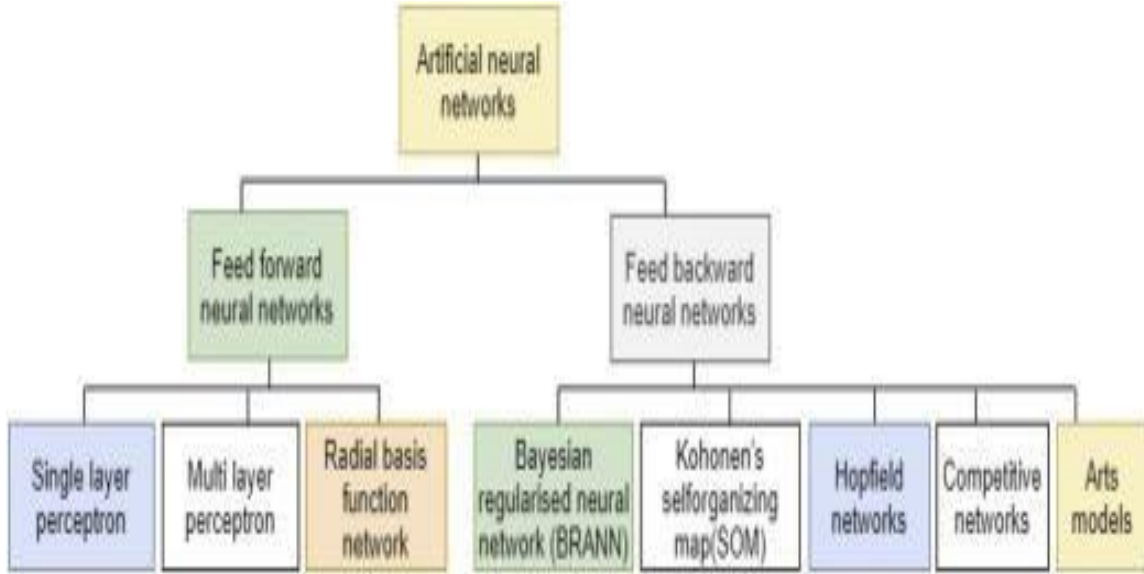
Şekil 1.8 AI Teknolojisi [89]

AI gelişimi incelendiğinde, beyin bilimi ile AI arasındaki bağlantıların önemli bir bileşke ile ilerlediği belirtilebilir [25].



Şekil 1.9 Yapay Zekâ Geliştirme [103]

AI biliminde öncü olan birçok bilim insanı aynı anda beyin bilimi ile de ilgilenmektedir [25, 28].

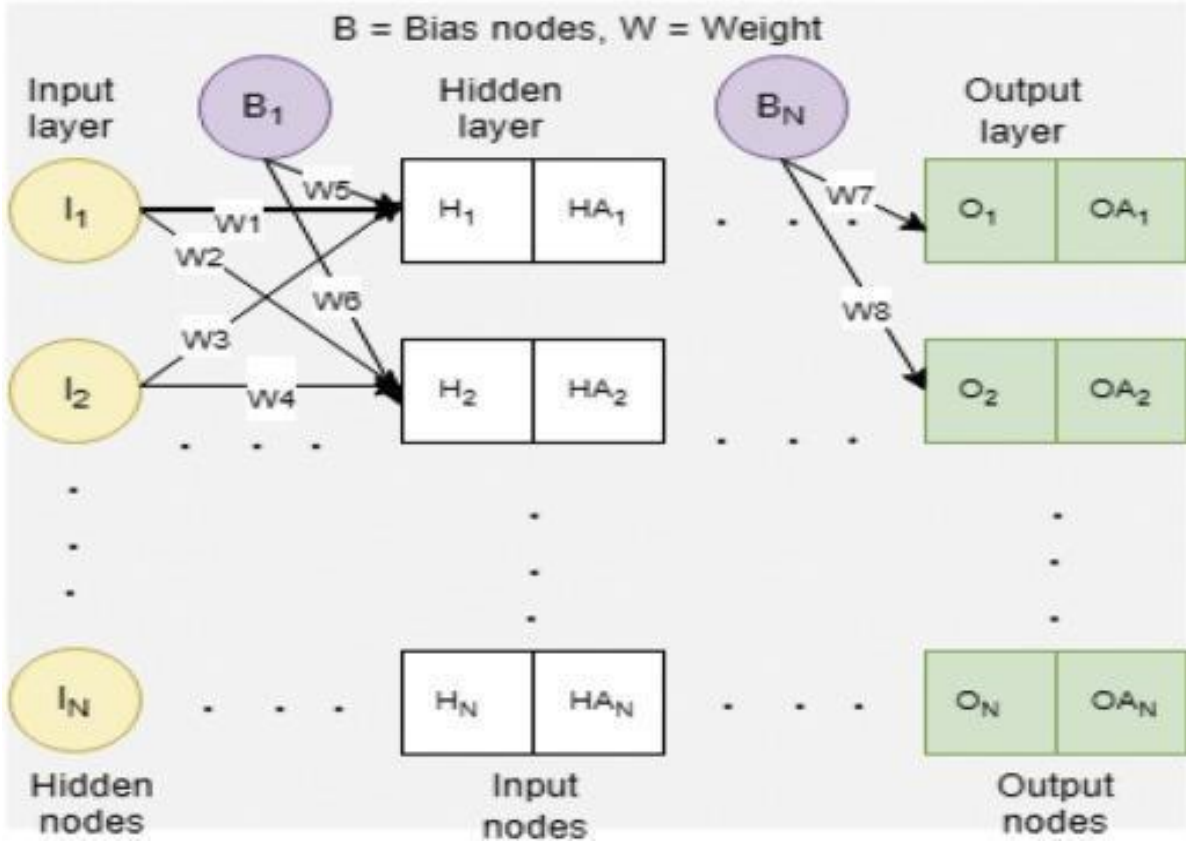


Şekil 1.10 Yapay Sinir Ağı Sınıflandırması [103]

Keşfi, mikroskopların kullanılmasıyla başlayan insan beyinde yer alan sinirsel bağlantıların, yapay sinir ağının oluşumunda ilham oluşturduğu bilinmektedir [29].

Özellikle, FFNN (Feedforward neural network), FITNET (Function fitting neural network), CFNN (Graded forward neural network) ve GRNN (Generalized regression neural network) gibi ANN (Artificial neural network) algoritmalarının kullanıldığı yapılan çalışmalar sunulmaktadır.

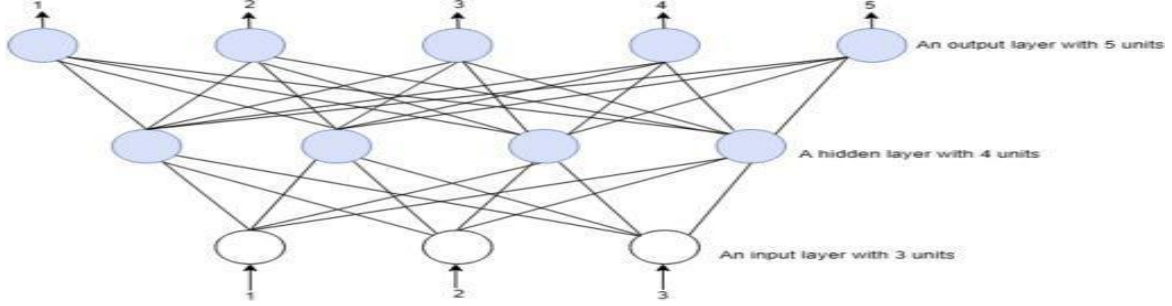
Yapay sinir ağı kapsamında, öznelik çıkarma yöntemi, büyük bir veri kümesi üzerinde eğitilen bir ağı belirlenen özelliklerinin öğrenilmesi sonucunda, yeni bir problem için, son katmanlar dışındaki tüm parametrelerin dondurulup; yalnızca onları yeniden eğiterek uygulanmasının bir göstergesidir [29].



Şekil 1.11 Tipik Bir Sinir Ağı Mimarisi [103]

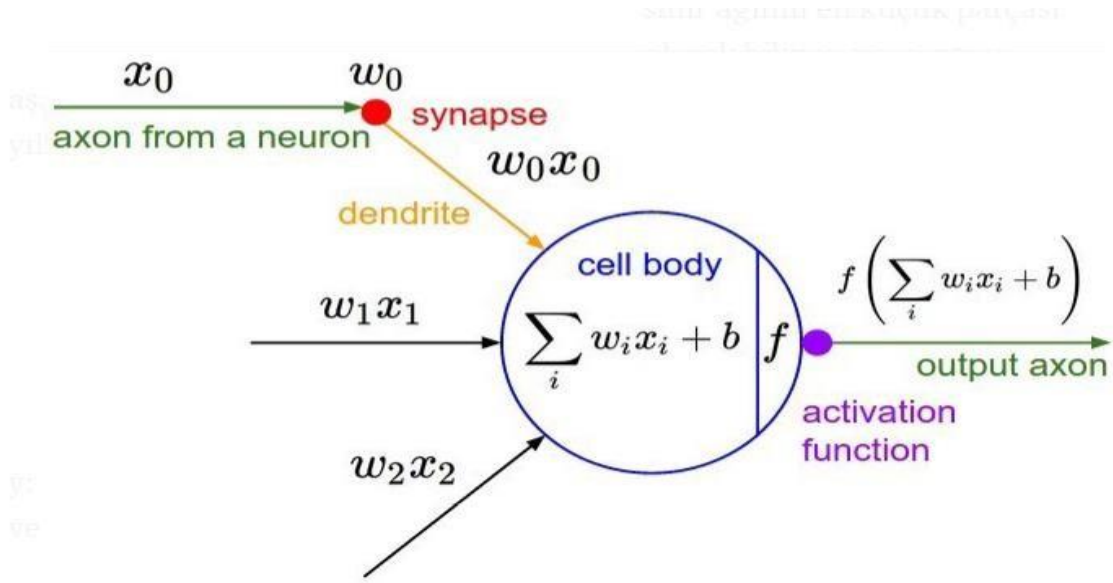
Alan Turing ve birçok yapay zekâ liderlerinden Marvin Minsky, Seymour Papert, John McCarthy ve Geoffrey Hinton iki alan da uzmanlaşarak; beyin bilimindeki sağlam çalışma birikimleri sayesinde AI'nın ilerlemesi boyutunda büyük katkıda bulundular[25-31].

Beyinde yer alan nöronlar, zihinsel aktivitelerinin altında yatan karmaşık devreler oluşturmak için birbirleriyle bağlantı kurarlar [29].



Şekil 1.12 İki Katmanlı İleri Beslemeli Ağ Mimarisi [103]

Sözde bir bağlantı şeması oluşturmak için bu bağlantıları haritalayıp, beynin nasıl çalıştığını öğrenmek son derece önemli bir adımdır [25, 29].

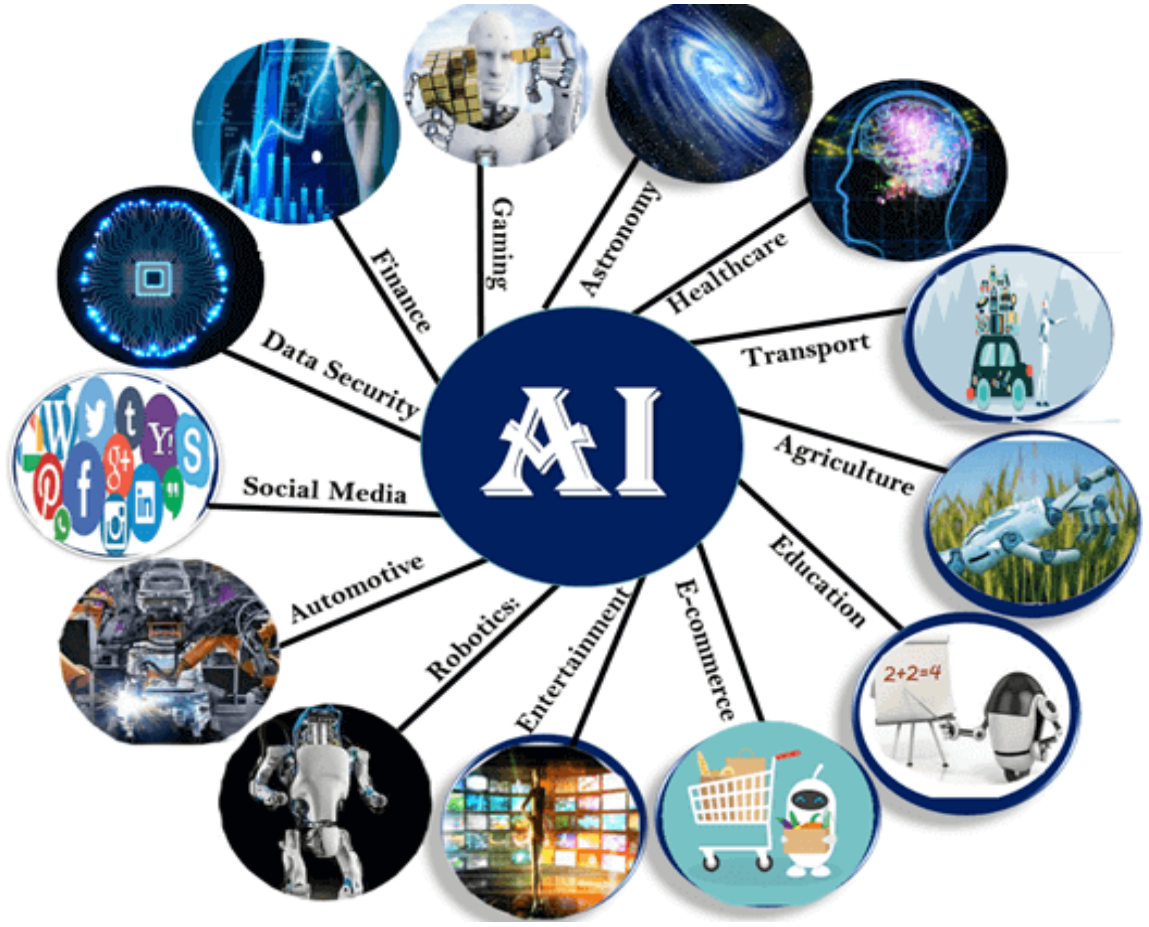


Şekil 1.13 Bir sinir hücresinin matematiksel gösterimi [104]

Bunu tam olarak yapmanın tek yolu şu an için elektron mikroskobu kullanmaktır. Bununla beraber, bu teknik o kadar zaman alır ki, en küçük doku örneklerini görüntülemek için tipik olarak binlerce saatlik çalışma gerekebilir [24-32, 71-79].

1.2.2.AI 'nın amacı

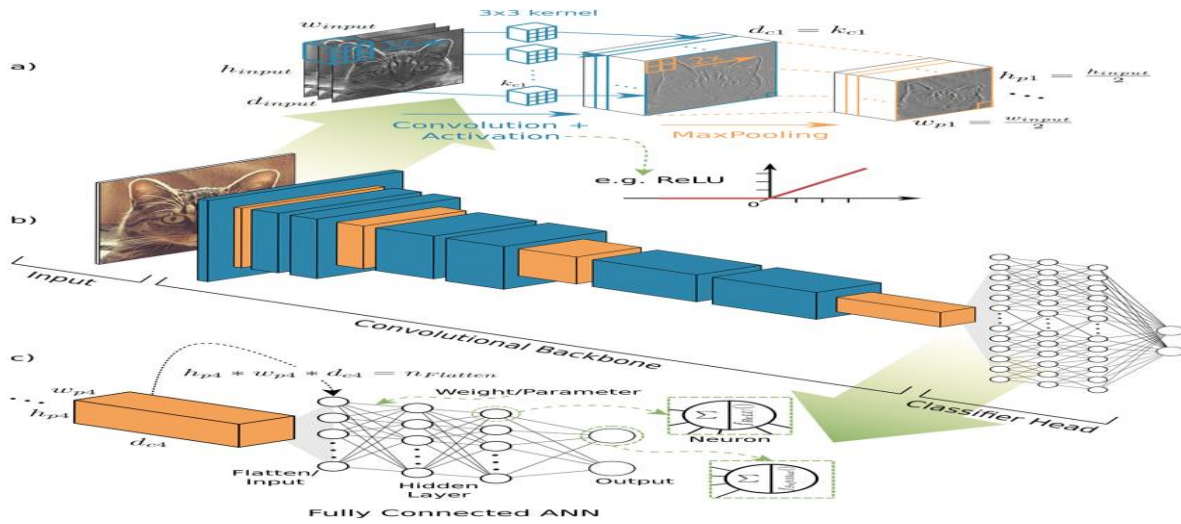
AI'nın amacı, algılayıcılar, tanıma, karar verme ve kontrol gibi işlevlerle; alt dallarının oluşumda önemli olan biyolojik veya insan zekâsı gerektiren görevleri gerçekleştirebilen teorileri araştırıp; bilgisayar sistemleri geliştiren ve bunların malzeme verileriyle işlenebilmesini sağlayan çok yönlü bir entegrasyon sürecidir [35].



Şekil 1.14 AI Uygulamaları [101]

1.2.3.ML(Machine Learning) nedir?

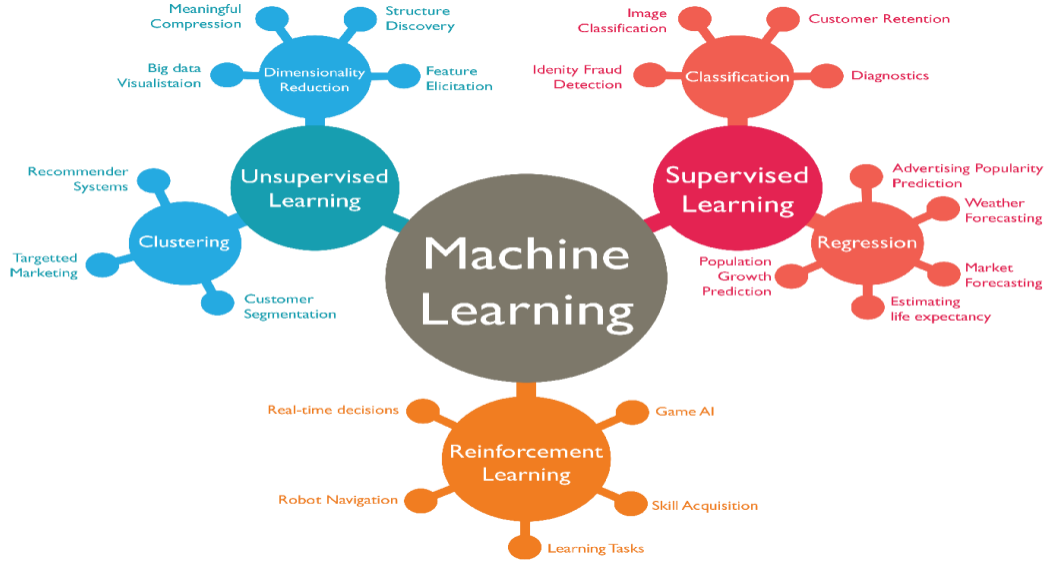
Son zamanlarda, ML teknikleri özellikle ölçeklenebilir olması, geniş algoritma sağlama ve uygulama yelpazesinin kapsamının genişliği sayesinde, belirgin bir şekilde kullanımı her alanda belirgin bir şekilde giderek yaygınlaşmaktadır. Bununla birlikte, nanoteknoloji ile ilgili verilerin uygulama alanı kapsamındaki ML'nin gerekliliği, insan öğrenme işinin bilgisayarlar tarafından gerçekleştirilmesi olarak açıklanabilir. Bu öğrenme dizininde çeşitli yaklaşımlar ve algoritmalar kullanılabilir [25-35].



Şekil 1.15 ML içeriğinde yer alan ANN ve CNN Ağının Şematik Gösterimi [105]

ML/AI teknoloji boyutunda, ML/AI araçlarının geliştirilebilmesiyle, mühendislik tasarım sistemlerinin gelecekteki verilerin, bilişimin ve görselleştirme yönlerinin önemli bir kısmını oluşturabilir. Muhtemelen, veri modeli yapısının ML/AI için ihtiyaç duyulan veri düzeylerini sağlamak üzere bilinçli bir şekilde tasarlanacak yöntemlerin geliştirilmesi gerekecektir. Burada özellikle, Jordon ve Mitchell; makine öğrenimi için "muhtemelen 21. yüzyılın en dönüştürücü teknolojilerinden biri olacak", ifadeleriyle birlikte devamında "bu nedenle mühendislik uygulamalarının herhangi bir uzun vadeli gelişiminde ihmal edilemez.", düşüncesiyle makine öğreniminin giderek bilimsel çalışmalarla bir arada ilerleyerek, insanlık için yapılan üretim sisteminin vazgeçilmez bir parçası olacağını belirtmektedir [35,36].

ML tekniklerinden, Denetimli öğrenme; algoritmaların beslendiği verilerin mantığına karar vermesine yardımcı olmak için "etiketli" olduğu zamanı gösterir. Denetimsiz öğrenme ise ham veri dışında herhangi bir girdi olmadan korelasyon bulmaya çalışan algoritma türüdür [35-40].



Şekil 1.16 Makine öğrenimi teknikleri: Denetimli Öğrenme, Denetimsiz Öğrenme [89]

1.2.4.DL(Deep Learning) nedir?

DL, son zamanlarda çok fazla dikkat çekici çalışmalarda yer alması; inanılmaz bir hızla tanıtılan çok boyutlu algoritmik ilerlemelere yol açan, ayrıca yoğun bir şekilde büyüyen bir alan olarak gerek bilim dünyasında gerekse dijital dönüşüm sistemlerinde yer bulmaktadır [41-42].

DL, nöronların canlılarda iletişim kurmasından, bu verilerden öğrenmesine dayanan; biyolojik olarak ilham alan bir öğrenme sürecinde var olan, bilgisayarların; gürültülü ve karmaşık veri kümelerinden kalıpların tespit edilebilmesi ve tahminlerde bulunabilmesi şeklinde sayısız niteliği içerisinde barındırmaktadır [43-46].

Şu anda derin sinir ağlarının, dili dönüştürmesi, görüntüleri sınıflandırma çeşitliliği ve görüntü oluşturma problemlerine ihtiyaç dahilinde cevap verebilmesiyle, çeşitli görevleri çözmek konusunda en başarılı makine öğrenimi tekniği olduğu söylenebilir. Bu tür modellerin birkaç eksiğinden bahsetmek gerekirse, insanlardan

farklı olarak, aynı anda birçok görevi sırayla öğrenememeleri kapsam boyutunda değerlendirilebilir [43-46].

DL, birden fazla işleme katmanından oluşan hesaplama modellerinin, yine birden fazla soyutlama düzeyiyle verilerin temsillerini öğrenmesine olanak tanınması; bu yöntemlerin, özellikle ilaç keşfi ve genomik gibi diğer birçok alanda en son teknolojiyi önemli ölçüde geliştirdiği çalışmaların sonuçlarında yer almaktadır [41,46].

DL'nin önemli bir kilit yönü olarak, içeriğindeki özellik katmanlarının insan mühendisler tarafından tasarlanmayıp; genel amaçlı olarak oluşturulan ve öğrenme prosedürü kullanılarak verilerden öğrendiğinin bilinmesidir. DL'nin yakın gelecekte çok daha fazla başarıya sahip olacağı düşünülmesinin nedenlerinden biri, bu alanda mühendislik uygulamalarında gereken çalışma verilerinin manuel gerekliliğini zamanla ortadan kaldırmasıdır. Bunun önemli sebeplerinden biri, mevcut hesaplama ve veri miktarındaki artışlardan kolay bir şekilde yararlanabilmektir. Son zamanlarda derin sinir ağları için geliştirilmekte olan yeni öğrenme algoritmaları ve mimarilerinin bu ilerlemeye hız kazandıracığı düşünülmektedir. DL teknikleri, derin malzeme bilimi olarak adlandırdığımız malzeme bilimi uygulamaları için de giderek artan bir ivme ile başarılı olarak kullanılmaktadır [41-46].

Bununla birlikte genel olarak değerlendirildiğinde ve ML baz alınarak düşünüldüğünde elbette, DL'nin bazı avantajları ve sınırlamaları da bulunmaktadır:

Geleneksel ML yöntemlerine kıyasla DL'nin üç temel avantajı bulunmaktadır.

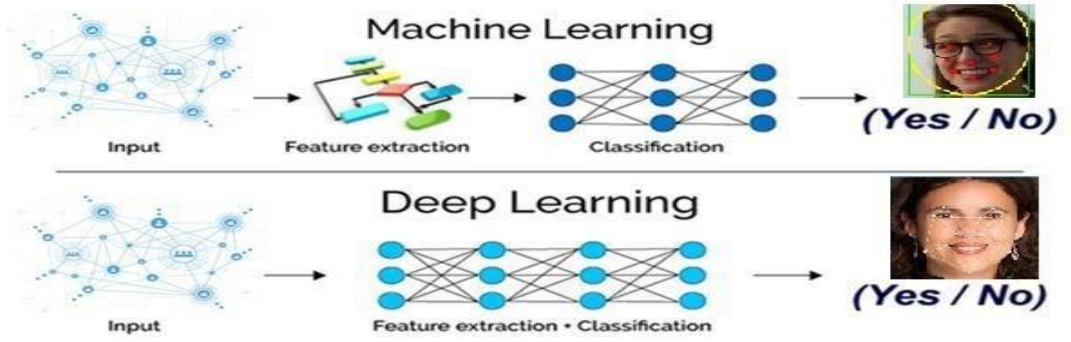
DL, büyük ölçüde özellik mühendisliği gerektirmemektedir [42-46]:

Genel anlamda bilimsel veya mühendislik uygulamaları düşünüldüğünde, iyi bir temsil dikkatli özellik mühendisliğini de içerisinde barındırıp; bu kapsamlı alanın bilgisinin yanı sıra uygun öznitelikleri bulmak için önemli manuel ve sezgisel çaba gerektirebilir. DL'de ise eğitim verilerinden ilgili özellikleri hiyerarşik bir şekilde otomatik olarak çıkarma yeteneği mevcuttur, böylece özellik mühendisliği çabasını azaltabilir ya da ortadan kaldırabilir. Bu, yalnızca öznitelikleri bulmak için harcanan manuel çabadan tasarruf edilmesine yardımcı olmakla kalmaz, aynı zamanda yeni bilgilerin önünün açılmasında da önemli bir etkiye sahip olabilir [42, 44-46].

- DL genel kapsamda büyük verilerle daha doğru sonuçlar ortaya çıkarır.
- DL daha hızlı tahminler üretebilme kapasitesine sahiptir [41, 46].

Bunlarla birlikte DL'nin uygulama aşamasında dört ana zorluk mevcuttur:

- DL büyük veri gerektirir: malzeme bilimi gibi birçok bilim ve mühendislik alanında hala nadirdir.
- DL büyük bilgi işlem gerektirir.
- DL ağ mimarisi arařtırmaları ve
- Model yorumlanabilirliđi başlıca sınırlılıkları olarak sıralanabilir [42, 46].



Şekil 1.17 Makine Öğrenmesi tekniklerinden Derin Öğrenme ile yüz tanıma örneđi [106].

2. LİTERATÜR

AI'nın alt dalı olan ML metotunun bir çeşiti olan DL mimarilerine bir önceki bölümde değinilmektedir. Bu bölümde deneysel kısmın alt yapısını oluşturan literatür taraması ile, CNN(Convolutional Neural Network) kullanılan kütüphanelere ve fonksiyonlara kısaca yer verilip; ardından araştırma probleminde kullanılan nanomalzemelerin her biri için özellikle işlerliği boyutunda açıklamalarına değinilerek; kullanım alanlarına kısaca yer verilerek; kapsam içerisinde; özellikle nanomühendislik, ileri teknolojilerinin oluşumunda, yapılan sınıflandırma ile ileride üretim modelleri için detaylı olarak kodlama yardımıyla uygulamada geliştirilmesi ve bilim dünyasında yer bulması açısından önem kazanması, gelecek çalışmalara işlevsellik katması açısından önem taşımaktadır [46-57].

2.1.CNN (Convolutional Neural Network)

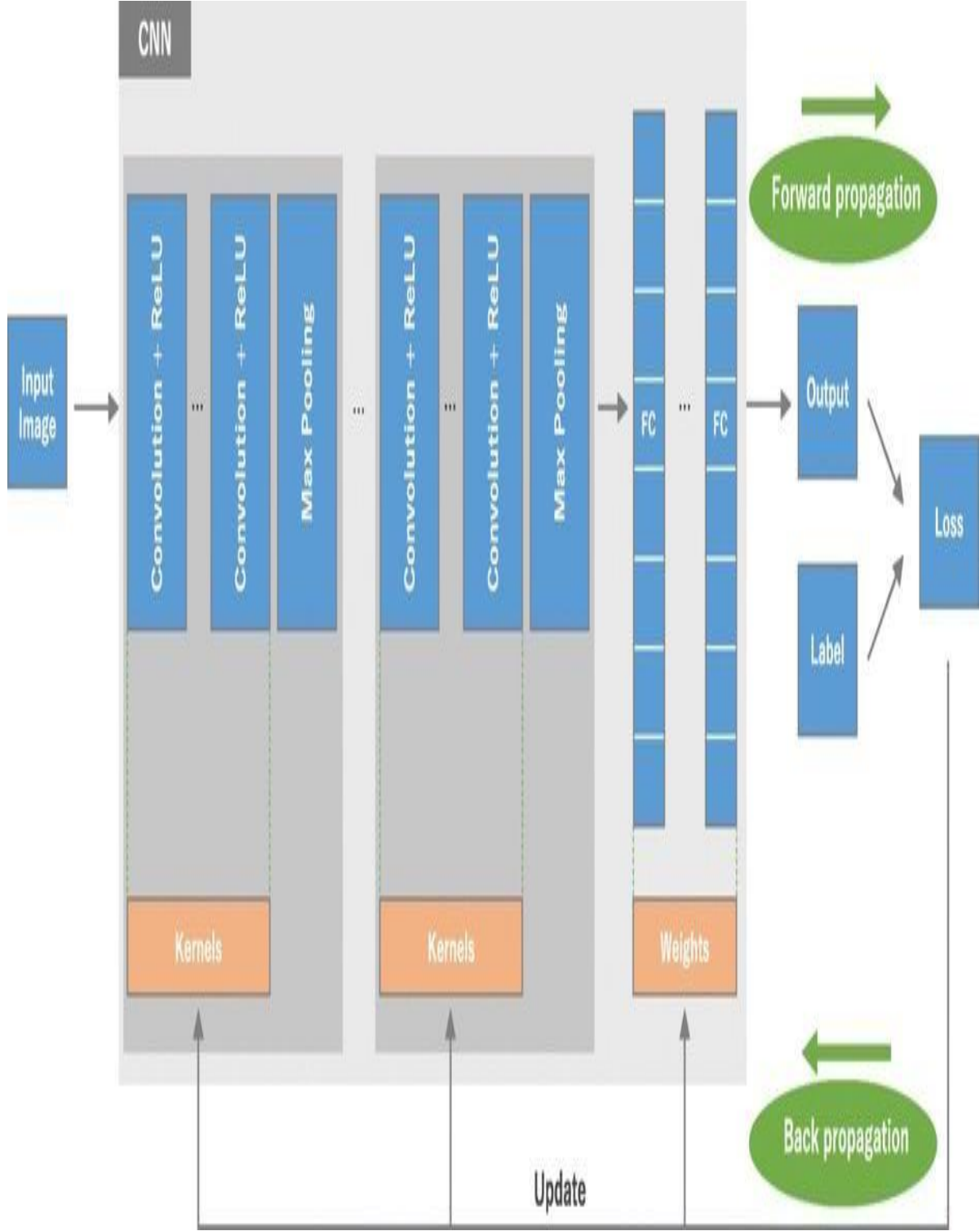
CNN, en yaygın olarak görüntü sınıflandırma ve tanıma için uygulanan derin sinir ağları alt kategorisidir. Giriş ve çıkış arasında CNN mimarisini oluşturmak için genelde üç tip katman kullanılır. Bu katmanlar; evrişim katmanı, havuz katmanı ve tam bağlantılı katmanlardır [49- 51].

Burada CNN' de yer alan evrişim katmanı, giriş verilerinin özelliklerini çıkarıp; gürültüyü azaltmak için kullanılır [50].

Havuzlama katmanı, girdi verilerini alt örneklerle alıp; her bölge için ortalama işlev veya maksimum işlevleri uygulamak için girişi küçük bölgelere böler.

Detaylı olarak yer vermek gerekirse, temel girdi/veri hattı için üç ana işlem gerçekleştirilir:

1. Veri kümesi içindeki verilerin ön işlenmesi,
2. Veri kümesinin karıştırılması,
3. Veri kümesinin içindeki toplu kümelerdir [50].



Şekil 2.1 CNN ağı Görseli [107]

Tez kapsamında yer alan deneysel çalışmada kullanılan CNN sınıflandırma için kullanılan; python kütüphaneleri aşağıda kısaca açıklanmaktadır [50, 57].

2.1.1.Tensorflow

ML/DL modellerinin uygulanması için eğitimi ve dağıtımı gerçekleştiren açık kaynaklı bir platformdur [52].

2.1.2.Keras

Hem CPU’larda hem de GPU’larda sinir ağı mimarilerinin uygulanması için kullanılan açık kaynaklı bir kitaplıktır [56].

2.1.3.Matplotlib

Özellikle yapılan çalışmaların görselleştirilmesi için etkileşimli çizelgeleri ve görüntüleri göstermek için kullanılan bir python aracıdır [57].

2.1.4.Eğitim (Train) veri kümesi

Sinir ağını doğrudan eğitmek için kullanılan bir veri kümesinin grubudur. Eğitim verileri (Train Data Sets), eğitim sırasında sinir ağına maruz kalan veri kümesini ifade edip; python kod sisteminde manuel olarak veya python kodu ile de yapılabilir [56,57].

2.1.5.Doğrulama (Validation) veri kümesi

Veri kümesi grubu, eğitim sırasında ağın performansını çeşitli yöntemlerle değerlendirmek amacıyla kullanılır [56,57].

2.1.6.Test (Test) veri kümesi

Veri kümesinin bu bölümü, eğitim aşamasının tamamlanmasının ardından ağımızın performansını değerlendirmek amacıyla oluşturulur [56,57].

2.2.Deneysel Olarak Kullanılan 10 Veri setinin Nanoteknolojideki Yeri ve Önemi

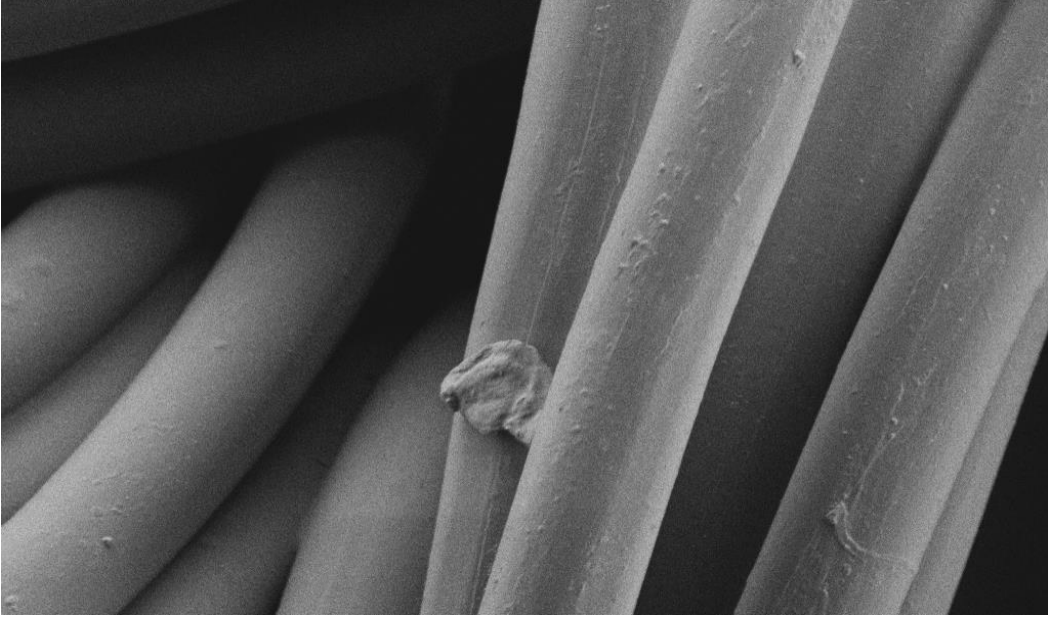
Nanomalzemelerin önemi üretilen sistemin bir parçası olarak önem taşımakta ve malzemelerin makro, mikro ve nano boyutta farklı özellikler sergilediği yapılan bilimsel çalışmalar kapsamında bilinmektedir.

Bu amaçla aşağıda deneysel çalışmaların veri kümelerini oluşturan malzemelerin neler olduğu açıklanarak uygulama alanları belirtilmektedir [58- 71, 94].

2.2.1Lifler (Fibres)

Elektron Dağılım Spektroskopisi (EDS) ölçümlerinin ve SEM mikrograflarının eş zamanlı değerlendirmesi, tekstil malzemelerinin, yüksek performanslı olup; fonksiyonel lif malzemeleri hakkında değerli bilgiler verilmesini olanaklı kılmaktadır. Modern dünyamızın, yüksek performanslı ve işlevsel malzemelere olan talep artışı gösterdiği gerçeği, bu durumun gelecekte de artacağını kanıtlamaktadır. Bunlara ek olarak, belirli yüksek performanslı malzemenin olumlu bakış açısını önemli ölçüde

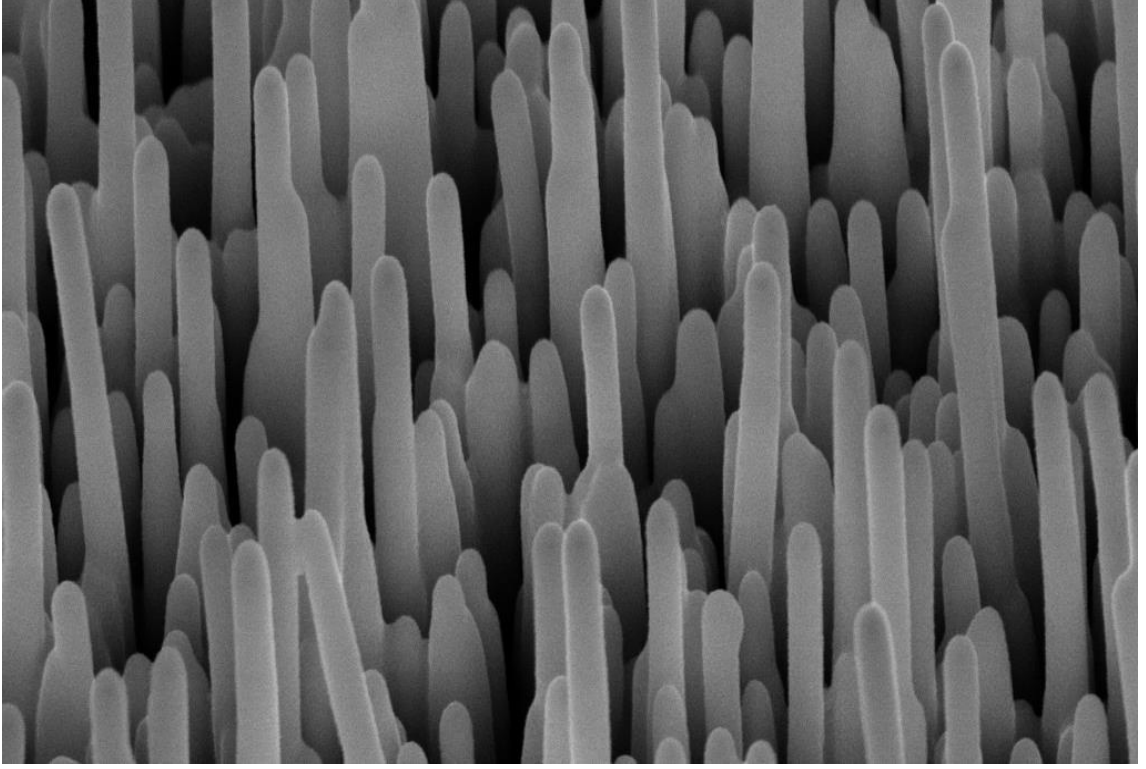
etkileyebileceği üç konu dikkate değerdir. Bunlardan ilki, yüksek performans gösteren ve fonksiyonel malzemelerin maliyetleri boyutunda, geleneksel malzemelere kıyasla önemli ölçüde daha yüksek bir önceliktir. İkinci olarak, yüksek performanslı malzemelerin avantajlı özellikleriyle eş zamanlı olarak düşünüldüğünde, bu malzemelerin her biri aynı zamanda bazı dezavantajlı özelliklere de sahiptir. Son olarak ve belki de en dikkate değer kısım, malzeme özellikleri mükemmel olsa bile, belirli fiber malzemelerin kullanımı ölçüsünde yasal düzenlemelerle kısıtlanabilir. Ayrıca fiberin tanımlanmasıyla, kaynakları hakkında önemli bilgiler de verebilir. Bununla beraber, fiber incelemenin asıl kısmı olan, iz kalıntı kapsamında ortaya çıkar. Detaylı bir örnek ile inceleme yapılan, iz dağılımının yorumlanması ve iddia edilen suç mahallindeki en olası eylemlerle ilgisi olan, bulunan parçalardan tekstil ve giysilerin yeniden yapılandırılması, belirli elyaf türleri aracılığıyla üreticinin ve üretim zaman çerçevesinin belirlenmesini içeren fiber istihbarat çalışmasında yer bulabilir. Tekstil hasarlarının, fiziksel uyumların araştırılması ve düğümlerin yeniden oluşturulması da tekstil araştırmalarında önemli rol oynayabilir [58,59,71].



Şekil 2.2 Lif SEM Görüntü Örneği [71]

2.2.2. Nanofiberler (Nanowires)

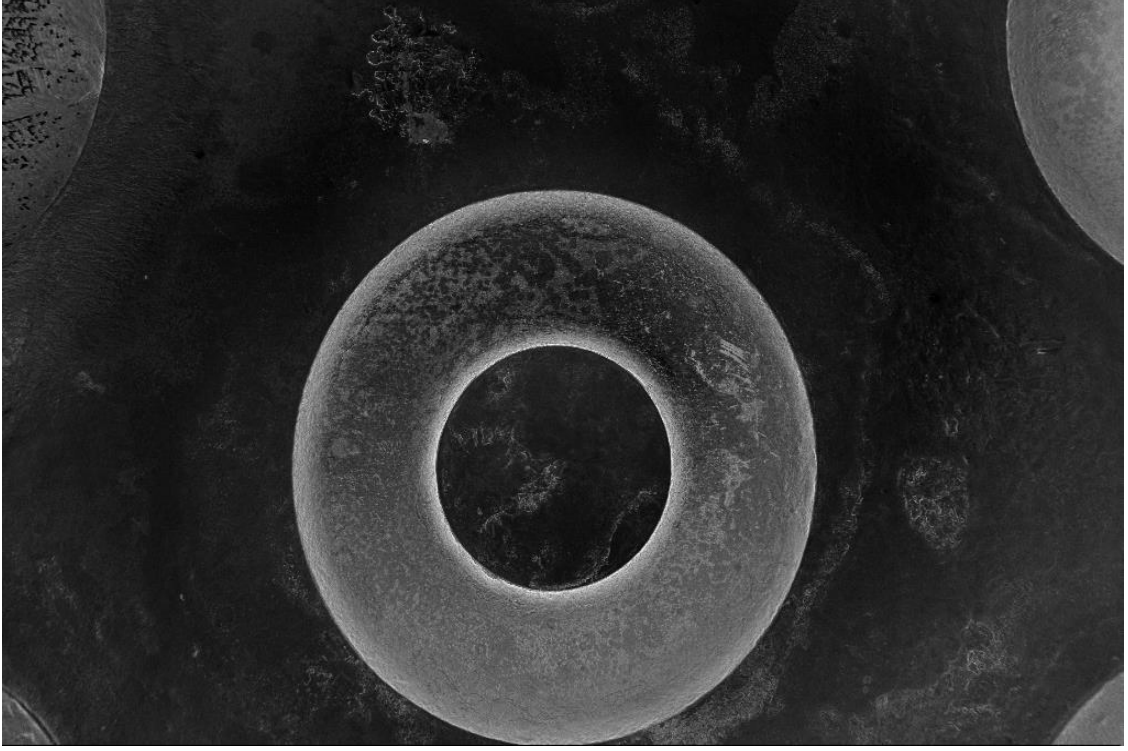
Enerji talebi, sanayi devriminin başlangıcından günümüze değin, hızlı ekonomik gelişmenin suni ivmesiyle gelen sürekli artışa katkıda bulunmaktadır. Yakıt tüketiminde, özellikle konvansiyonel fosil yakıtlar, bunun sonucu olarak enerji talebini karşılamak için sürekli olarak artış göstermektedir. Ayrıca nanoteknoloji alanındaki gelişmeler ölçüsünde, karbonu etkin bir şekilde yakalamak için tek boyutlu (1D) olan nanoyapılar oldukça detaylı bir araştırma döngüsünün içerisinde yer almaktadır. Nanorodlar, nanofiberler, nanoteller, nanosilindirler ve nanobıyıklar olmak üzere birkaç 1D nanoyapı bulunmaktadır. Her morfoloji için, karbon emisyonunu azaltmak için, onları diğer nanoparçacıklardan üstün kılan eşsiz bir karbon yakalama özelliği sergilemektedirler. Ayrıca karbon yakalama uygulaması ve karbon depolama kapsamıyla düşünüldüğünde, sürdürülebilir enerji üretimi ve kirlilik kontrolü yoluyla çevresel sürdürülebilirliğin sağlanmasında kullanılan farklı 1D nanoyapılar hakkında bilgiler vermesi açısından önem taşımaktadır [60, 71].



Şekil 2.3 Nanofiber SEM Görüntü Örneği [71]

2.2.3.Desen Yüzeyi (Patterned _surface)

Son zamanlarda, polimerik malzemelerin tribolojik özelliklerini değiştirmek amacıyla mikro ölçekli yüzey desenlerinden yararlanmak için yüksek hassasiyet gösteren üretim tekniklerinin geliştirilmesi yeni olanakları ortaya çıkarmıştır. Ayrıca, yüzey karakterizasyonuna etkisinin, malzeme mekanik parametreleriyle birlikte düşünüldüğünde, istenen özelliklere sahip yüzeylerin tasarımına izin vererek tam olarak anlaşılması gerekir. Yüzey desenleme yoluyla, yüzeylerin hem statik hem de dinamik sürtünme katsayılarını değiştirmenin mümkün olduğu bilinmekle birlikte, bu da ayarlanabilirlik sağlama olasılığını gösterir. Bunun sonucu olarak, tribolojik özelliklerin, aşırı ayarlanabilirliğine sahip yüzeyler elde etmek maksadıyla yüzey deseninin sürtünme etkilerinin tanımlanmasına ve anlaşılmasına daha fazla katkısı olduğu çalışmalar kapsamında değerlendirilebilir [61, 71].

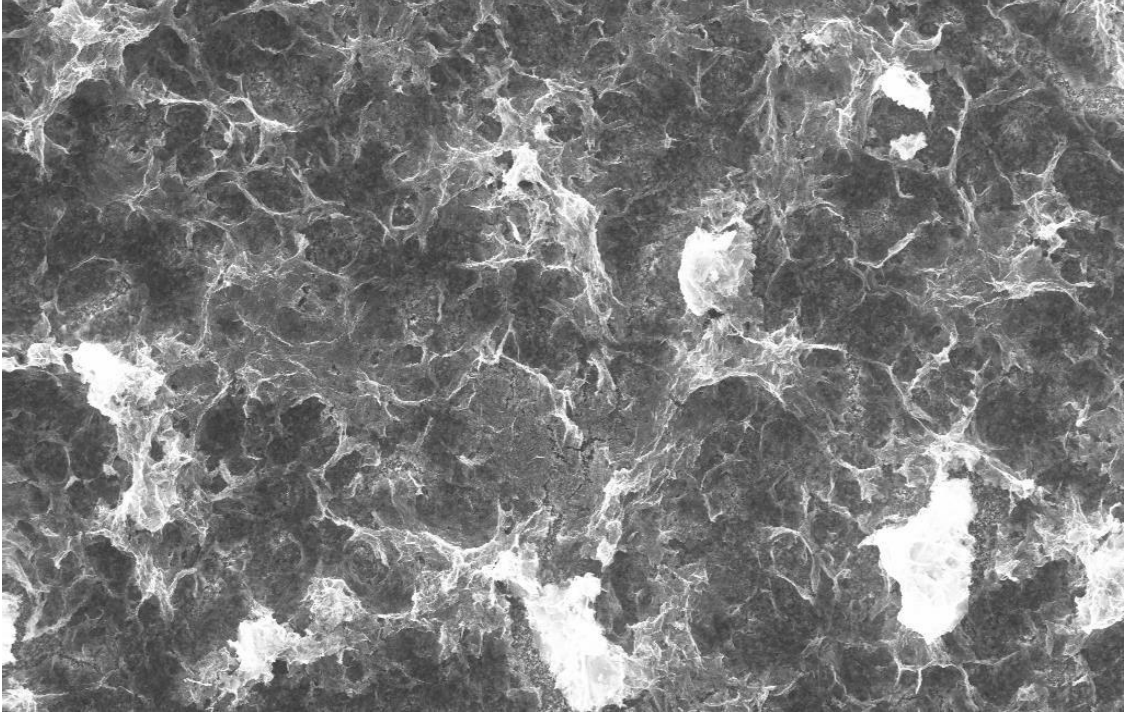


Şekil 2.4 Desen Yüzeyi SEM Görüntü Örneği [71]

2.2.4.Film Kaplı Yüzey (Films_Coated_Surface)

Tabletler, farmasötik endüstrileri tarafından üretilen en yaygın katı oral dozaj şekli ibarettir. Başarılı ve tutarlı tablet üretiminde film kaplı yüzey için gerekli olan çeşitli zorluklar bulunmaktadır.

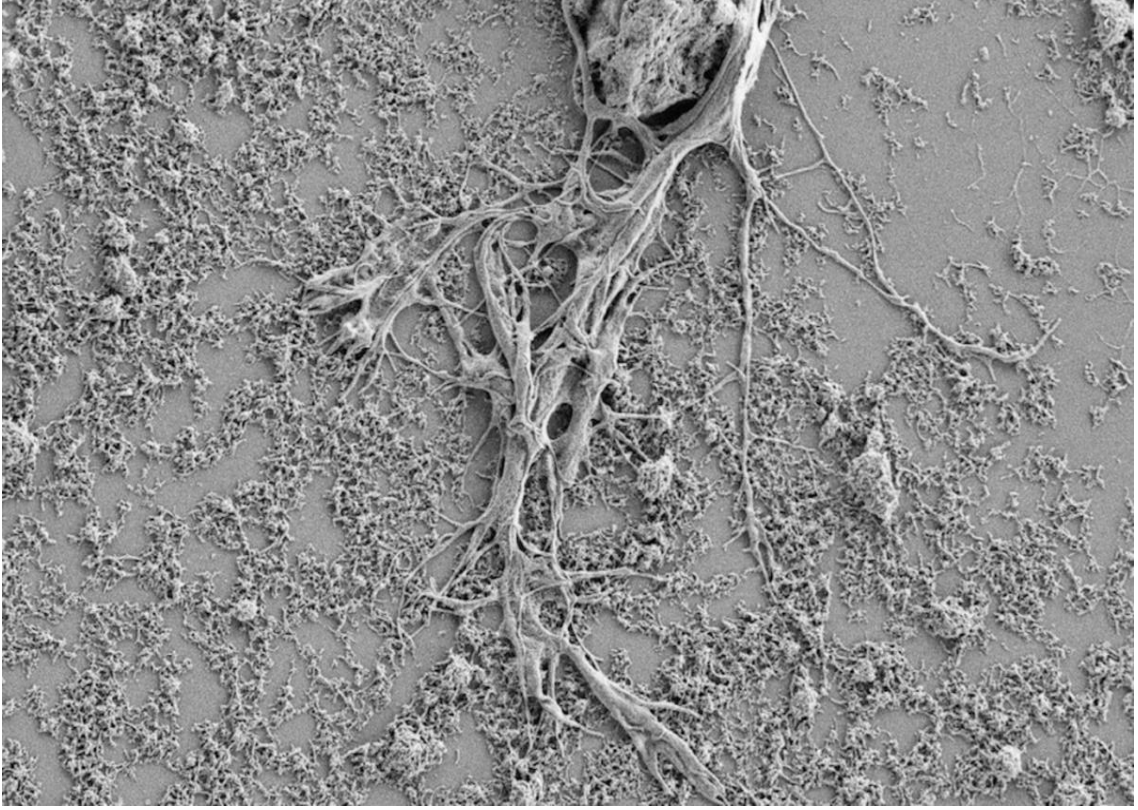
İyi bilinen bir kalite sorunuyla gelen, genellikle yetersiz fiziksel güç nedeniyle oluştuğu bilinen ve bunlara ek olarak işleme, paketlenme ya da nakliye sırasında kırılma veya aşınma nedeniyle; görünür yüzey kusurları dikkate değerdir. Yüzey mukavemetinin ve yüzey kusuru riskinin nicel olarak değerlendirilmesine izin veren tekniklerin, kalite kontrolüne büyük ölçüde yardımcı olacağı düşünülmektedir [62,71].



Şekil 2.5 Film Kaplı Yüzey SEM Görüntü Örneği [71]

2.2.5.Biyolojik (Biological)

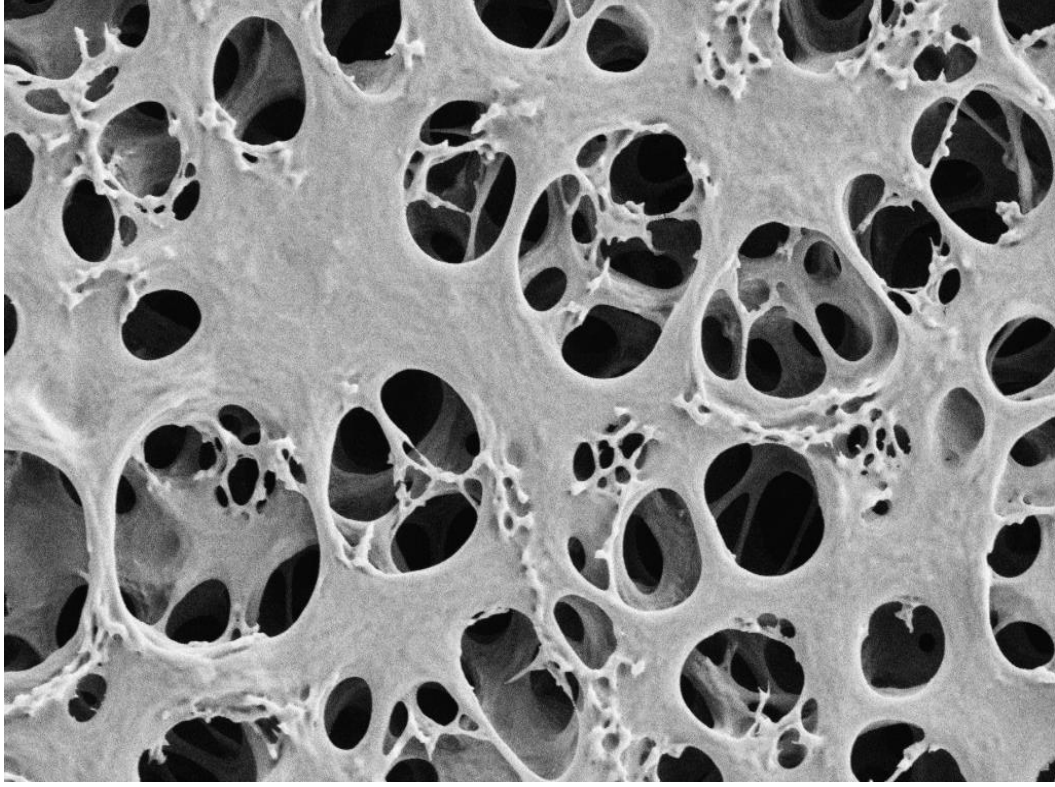
Biyolojik yapıyı keşfetme ve yapısal organizasyonu işlevsel önemle ilişkilendirme arayışında, bilim insanları çok çeşitli mikroskoplar geliştirdi. SEM, maddenin yapısal organizasyonunu araştırmak amacıyla faydalı araçların geliştirilmesi için yeni ve önemli bir ilerlemeyi temsil etmektedir. Hem teknoloji hem de metodolojideki son gelişmeler, bitki ve hayvan doku ve organlarındaki hücre içi detaylarla beraber, yüzeyi ortaya çıkarmak maksadıyla SEM' in özel olarak veya diğer mikroskop türleri ile bağlantılı olarak kullanıldığı sayısız biyolojik yayınlara sonuçları paylaşmaktadır [63, 71].



Şekil 2.6 Biyolojik SEM Görüntü Örneği [71]

2.2.6.Gözenekli Sünger (Porous_Sponge)

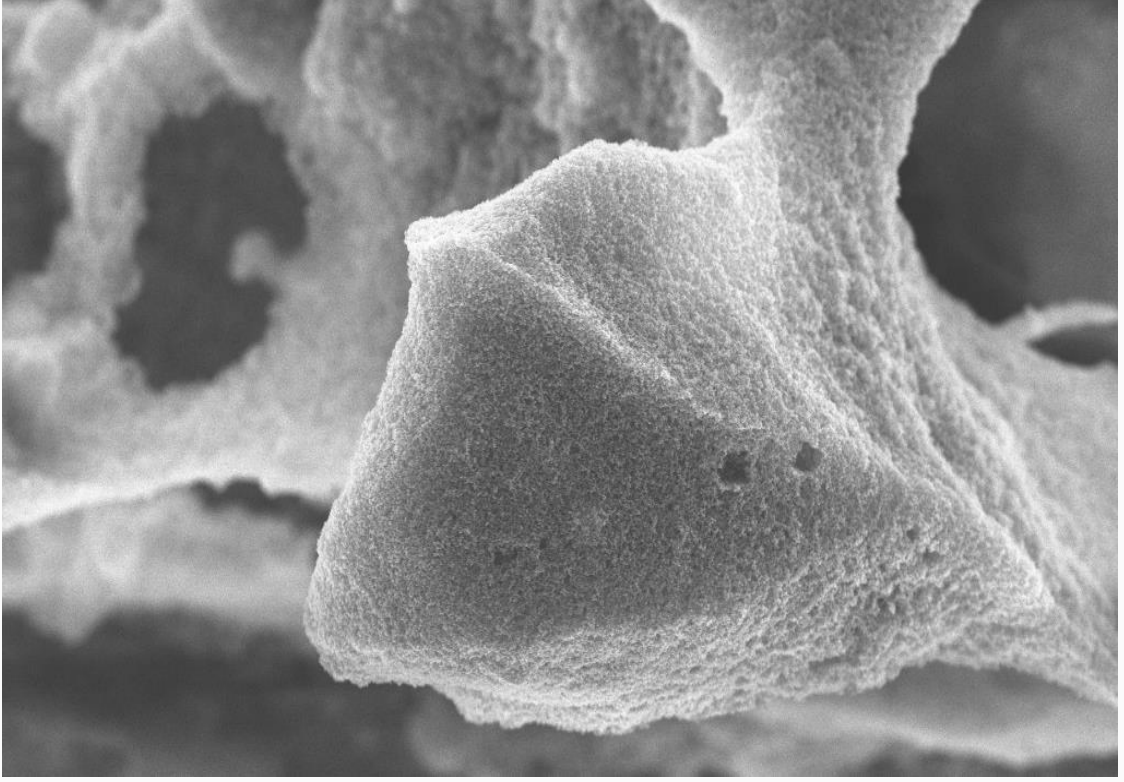
İpek fibroin gözenekli süngerlerinin, doku mühendisliği uygulamaları için literatürde geniş boyutta çalışıldığı, yıllar boyunca, söz konusu dokuya uyarlanması gereken özellikler açısından taleplerin çoğunu karşılamak için çeşitli üretim yöntemleri geliştirilmektedir. Önemli bir kullanım alanı olan yeni eşlenik biyo-bazlı ve biyolojik olarak parçalanabilen sünger, petrol sızıntılarını temizlemek ve su arıtmak için umut verici sürdürülebilir bir malzeme olarak önem taşımaktadır [64-65, 71].



Şekil 2.7 Gözenekli Sünger SEM Görüntü Örneği [71]

2.2.7.Toz (Powder)

Nihai katkı maddesiyle üretilen parçanın mekanik özelliklerinin, statik ve dinamiği olarak, yazdırılan parçanın mikro yapısını etkileyen çeşitli girdi ve parametreler tarafından yönetildiği bilinmektedir. Eklemeli üretim için kullanılan en yaygın sistemlerden (seçici lazer eritme ve elektron ışını eritme) hammadde olarak metal tozları kullanılmasıyla beraber, bu yapıların karakterizasyonu açısından önemlidir. Tozun kalitesi boyut, yüzey morfolojisi, bileşim ve iç gözenekliliğinin miktar göstergesidir. Eski ve geri dönüştürülmüş metal tozların, havaya maruz kaldıklarında oksitlenme ve nem toplama eğilimi taşımaktadır. Bu farkındalıkla, inşa edilen oda atmosferinin bir koruyucu gaz içermesi gerektiği bilinmektedir. Ancak bu gazlar kimyayı işlenebilirliği ve ısı transferi etkileşimleriyle; basılı parçanın mikro yapısındaki etkilerini ortaya çıkarmaktadır [66, 71].

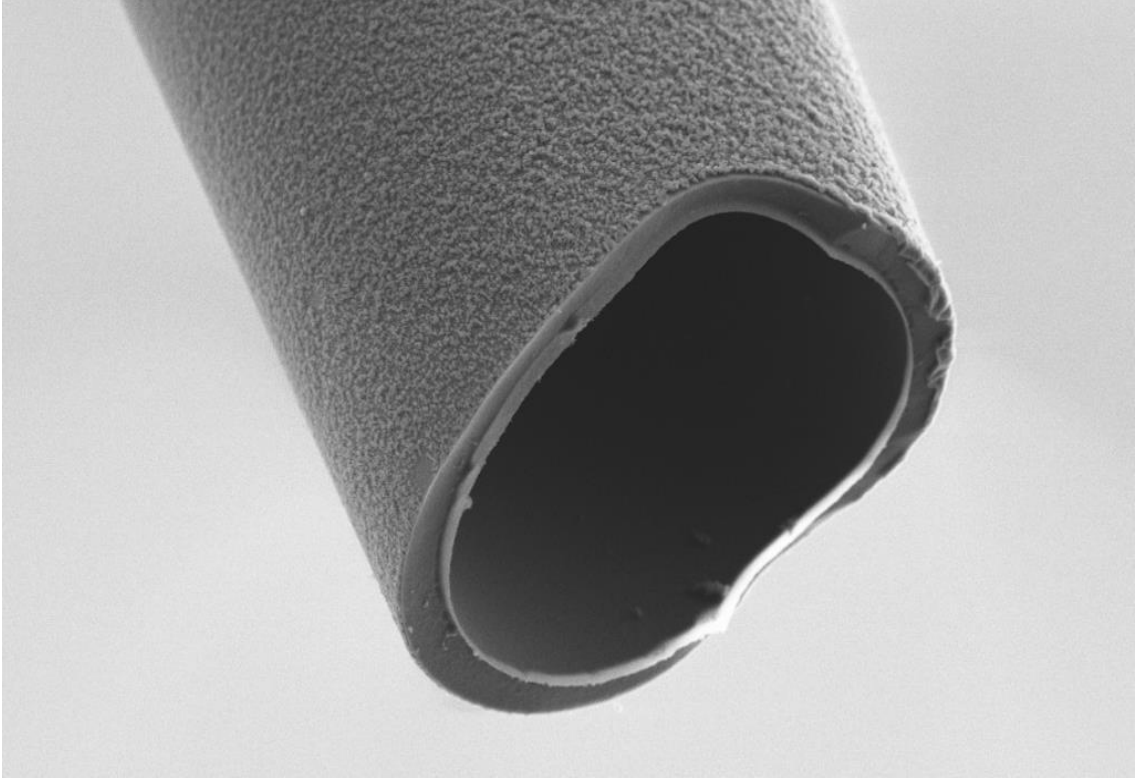


Şekil 2.8 Toz SEM Görüntü Örneği [71]

2.2.8.Uçlar (Tips)

Nanomanipülasyon, hata analizi ve kalite kontrolü için elektriksel karakterizasyon gerçekleştirmek üzere nanoyapılar üzerinde nanoprobaların otomatik olarak konumlandırılmasını sağlamaktadır.

Eşzamanlı görüntüleme ve nanomanipülasyon düşünüldüğünde, SEM'ler içinde nanorobotik manipülasyon laboratuvarlarının kurulmasında teknik ilerlemeler, nano cihaz prototiplerinin montajını ve hücre altı organellerinin analizini mümkün kılmaktadır. Son yirmi yılda kaydedilen önemli ilerlemelere rağmen, yüksek hassasiyet, sağlamlık, esneklik ve yüksek verim ile 3D nanomanipülasyonun gerçekleştirilmesinde zorluklar halen önemli ölçüde devam etmektedir. Programlanabilirlik, otomasyon ve özgüllüğün benzersiz avantajları bir arada düşünüldüğünde, bir SEM içindeki nanomanipülasyonu özellikle uçları kısmında bilimsel keşiflerin güçlü bir itici güç olarak hizmet etmeye devam edip; nano bilimler ve nanoteknoloji endüstrileri için daha güçlü bir işgücü teknolojisine dönüşmesinin mümkün kılınmaktadır[67,71].



Şekil 2.9 Uçlar SEM Görüntü Örneği [71]

2.2.9.Partikül (Particles)

Bir nanomalzemenin en temel ve hatta tanımlayıcı özelliklerinden biri, onu oluşturan bileşen boyutudur. Tipik olarak, böyle bir malzemenin herhangi bir fiziksel ya da kimyasal karakterizasyonu düşünüldüğünde, boyut ve boyut dağılımının doğru ölçümü ile başlayıp; şekli, yapısı, kristallik özelliği, kimyasal bileşimi, spesifik yüzey alanı ve yüzey kimyası gibi diğer ilgili parametrelerin belirlenmesi ile devam etmektedir. Bununla beraber, partikül boyutu gibi aldatıcı ölçüde basit bir derecelendirme bile, özellikle numune polidispers olduğunda ve nanopartikül morfolojisi ideal, küresel bir şekilden saptığında zorlu bir analitik görev oluşturabilir.

Mikroskobik partikül maddenin, çevresel ve mesleki maruziyetlerinin önemli bir bileşenidir. Bu tür maruziyetlerin bariz örnekleri şunlardır:

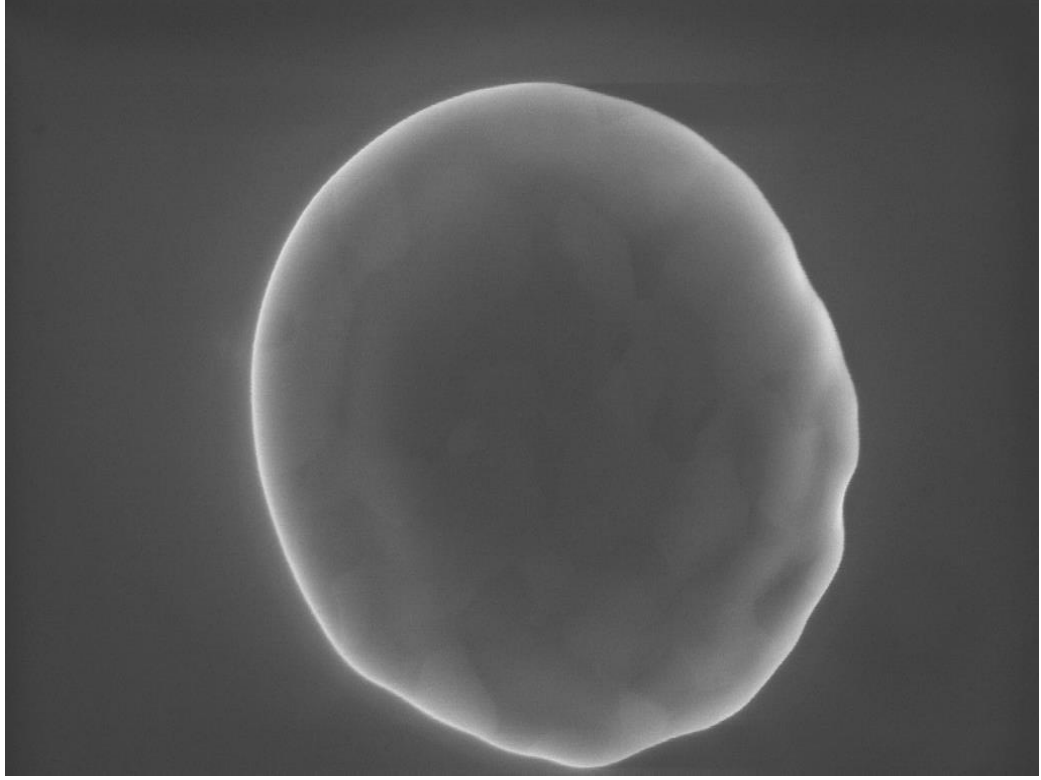
- Atmosferik aerosoller
- Kaçak tozlar
- Konsolide olmayan zeminler

- Askıda su kaynaklı tortular
- Endüstriyel süreç emisyonları şeklinde devam etmektedir.

Analitik elektron mikroskobu, diğer yöntemlerle kolayca karakterize edilemeyen partiküllü malzemelerin tanımlanmasında, karakterizasyonunu ve miktar tayinini kolaylaştıran yerleşik bir teknik olarak açıklanabilir.

Tekniğin kanıtlanmış bazı uygulamalarına aşağıda yer verilmektedir:

- Atmosferik parçacık kaynağı paylaşırma çalışmaları
- İçme suyunda bulanıklık kaynaklarının izlenmesi
- Kaçak izabe tozlarının tanımlanması
- Karasal çökellerde çökelme geçmişleri
- Deniz ve tatlı su çökellerinin analizi
- Ev tozunda kurşunun karakterizasyonu
- Endüstriyel tozların boyut dağılımı
- Akciğer tozu yükünün ölçülmesi
- Mahsul yüzeylelerinde partikül birikimi
- Toprak kurşun çözünürlük çalışmaları
- Buzdaki parçacıkların analizi
- Silah atış kalıntısı çalışmaları
- Dokulardaki yabancı maddelerin tanımlanması
- İç ve dış hava kirliliği çalışmaları
- Adli/tıbbi-hukuki soruşturmalar
- İşyeri/endüstriyel hijyen çalışmaları
- Pnömokonyoz araştırması (asbest, silika vb.) dir [68,69, 71].



Şekil 2.10 Partikül SEM Görüntü Örneği [71]

2.2.10.Mems cihazları ve elektrotları (MEMS_devices_and_electrodes)

Birçok boyutta, nanoteknoloji arařtırmaları MEMS cihazlarına baęlıdır. Aslında, nano ölçekli alan ile arayüz oluşturmak için çeşitli MEMS teknolojileri gerekli olduęu uygulamalarda kendini göstermektedir.

MEMS'in dięer optik uygulamalarına ařaęıda yer verilmektedir:

- Ayarlanabilir lazerler ve filtreler
- Deęişken optik zayıflatıcılar
- Optik Spektrometreler
- Barkod okuyucu
- Maskesiz litografidir.

Bununla beraber; MEMS'in dięer çeşitli uygulama örnekleri ařaęıda kısaca açıklanmaktadır:

-Mürekkep püskürtmeli yazıcılar için MEMS püskürtme uçları ve pompaları (mikroakışkanlar),

-İletişim uygulamalarında kullanılan anahtarlar, ayarlanabilir kapasitörler, faz kaydırıcı rezonatörler, indüktörler, filtreler ve deęişken antenler gibi radyo frekansı

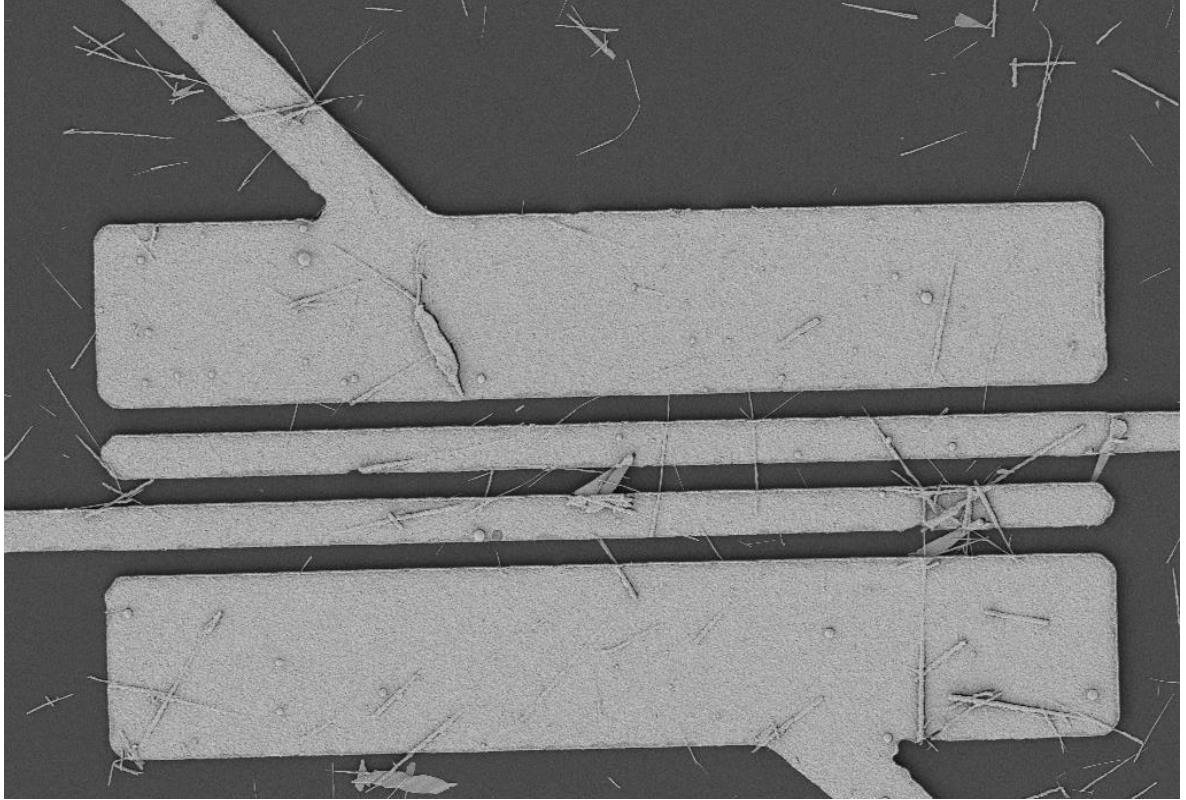
(RF) cihazları,

-İtici hareketini (veya slosh) kontrol edebilen uçak ve uzay araçları için yakıt dağıtım sistemleri,

-Mikro yakalayıcılar ve çeşitli alanlarda kullanılan kıskaçlar ya da cımbızlar; mikrondan nano boyutlu bileşenleri tutturmak, almak ve taşımak için,

-Mikrofonlar (ağırlıklı olarak akıllı telefonlarda ve tabletlerde kullanılır ve tabii ki casus yazılımlarda),

MEMS kablosuz ve karışık sinyal cihazları, biyo-çipler, güç cihazları ve optoelektronik gibi diğer bileşenlerle tek bir pakette entegre etme heterojen entegrasyonudur [70, 71].



Şekil 2.11 Mems Cihazları ve Elektrotları SEM Görüntü Örneği [71]

3.YÖNTEM

Görüntü sınıflandırma için, CNN 'de bir evrişim, kümeler arasında yer alan iki öge arasındaki bir nokta ile çarpımını tamamlayan, matematiksel bir ifadedir. DL boyutunda evrişim işlemi, evrişim katmanı içindeki filtre/çekirdek ve görüntü verisinin dizisi üstünde hareket etmektedir. Bu sebeple, bir evrişim katmanı, filtreler ve bir evrişimli sinir ağından geçen görüntüler arasında meydana gelen evrişim işlemi barındıran bir katman şeklinde düşünmek gerekir [48,49, 72,73, 74-107].

Deneysel çalışmada kodlama aşamasında kullanılan yöntem makine öğrenmesinin CNN sınıflandırma tekniğidir. CNN sınıflandırmada kullanılan fonksiyonların açıklamalarına aşağıda yer verilmektedir [72,73].

Aktivasyon Fonksiyonu (Activation Function): İşlem yapılan nöronların çıktısını ya da sinyallerini normalleştirilmiş bir sonuca dönüştüren matematiksel bir detaydır. Bir sinir ağının bir bileşeni olarak bir aktivasyon fonksiyonunun ana hedefi, ağ içinde doğrusal olmamayı sağlayıp; bununla beraber bir aktivasyon fonksiyonunun dahil edilmesi gereken sinir ağının daha fazla temsil gücüne sahip olmasını ve karmaşık fonksiyonların çözümlemesini sağlar [74].

RELU (Rectified Linear Unit Activation Function):Bir nöronun değer sonuçlarını dönüştüren bir tür aktivasyon fonksiyon türüdür. RELU ile bir nörondan alınan değerlerin uygulanan dönüşüm için, $y = \max(0,x)$ formülüyle temsil edilmesi gerekir. RELU etkinleştirme işlemine değinecek olunursa, nörondan gelen tüm negatif değerleri 0'a limitler ve pozitif değerlerin değişmeden kalmasını sağlar. Bu matematiksel dönüşümün sonucu olarak mevcut katmanın çıktısı olarak kullanılan bir sinir ağı içinde ardışık bir katmana girdi olarak kullanılmasını hedefler [73,74].

Softmax aktivasyon fonksiyonu (Softmax Activation Function): Bir girdi vektörü içindeki bir dizi sayının olasılık dağılımını bitirmek amacıyla kullanılan bir aktivasyon fonksiyonu tipidir. Bir softmax etkinleştirme fonksiyonunun çıktısı, içindeki değerler kümesinin bir sınıf veya olayın meydana gelme olasılığını temsil ettiği bir vektördür. Vektör içindeki değerlerin tümü 1'e eşit olmalıdır [74,75].

Bırakma (Dropout): Bırakma tekniği, bir sinir ağı içindeki birbirine bağlı nöronların sayısını rastgele azaltarak çalışmasını sağlar. Bunlara ek olarak eğitimin her adımında, nöronun bağlantılı olduğu nöronların harmanlanarak katkılarından çıkarılma

şansı oluşmasını sağlar [76].

Toplu Normalleştirme Katmanı (Batch Normalization Layer): Toplu Normalleştirme de amaç, önceki katmandan gelen girdiler üzerinde işlemler gerçekleştirip ek bir katmanın eklenmesinden sonra bir sinir ağı içindeki kararsız gradyanların etkisini azaltan bir durumdur. İşlemler sırasında giriş değerlerinin standartlaştırılıp, normalleştirilmesiyle; giriş değerlerinin ölçeklenmesi ve kaydırma işlemleriyle dönüştürülmesine yarar [77].

Max Pooling Katmanı (MaxPooling Layer): Bir alt kategoride ki örnekleme katmanında, bir birimin alıcı alanına düştüğü piksel sayılarının maksimum piksel değerinin çıktı olarak alındığı bir alt örneklemedir [78].

Düzleştirme Katmanı (Flatten Layer): Bir giriş şekli almasıyla birlikte giriş görüntüsünün verilerini tek boyutlu bir dizi olarak düzleştirilmesini sağlar [79].

Yoğun Katman (Dense Layer): Yoğun bir katmanın içerisinde gömülü olan rastgele nöron içerip; her nöronun algılayıcı vurgusunu yapar [80].

4.BULGULAR

Bilindiği gibi AI'nın alt dallarından ML kullanımı, bilimsel çevrede hızla yaygınlaşmaktadır. ML, çok geniş bir perspektifte kullanılmaktadır.

Tez içeriğini oluşturan, bulguların oluşumunda; ML'nin, DL metodunda görüntü sınıflandırması yapmak için CNN yöntemi kullanılarak; SEM görüntüleri 10 veri setinin eğitimi gerçekleştirildi [25-31, 35-42]. Burada deneysel çalışmada kullanılan hazır veri setlerinin kaynağı, *“Trieste'deki CNR- IOM 'un TASC laboratuvarında bulunan ZEISS SUPRA 40 çözünürlüklü cihaz ile 100 bilim insanı tarafından 5 yıl boyunca elde edilen görüntülerin, toplam 22.000 SEM verisinden”* oluşturulduğu bilinmektedir. Tez kapsamında yapılan deneysel çalışmada veri setlerinden oluşturulan 10 kategorilik prototipin verileri tek tek incelenip; çözünürlük, görüntü boyutu ve kalitesine göre veri seçimi yapıldı [71].

Tekrar oluşturulan veriler için, dikkate alınan özellik, görüntü kalitesi olmakla birlikte; 10 veri seti oluşturulmasının ardından her bir veri seti için 100 veri elle manuel olarak seçilip, toplam 1000 veri üzerinde yapılan eğitimin tamamlanması sağlandı [58-70, 71].

Oluşturulan veri seti *“SEM10”* adıyla bilgisayar üzerine kaydedildikten sonra; veri setinin SEM10.zip dosya biçiminde bilgisayara kaydı yapıldı. Ardından görüntülerin eğitimi için, SEM10.zip dosyası Google Drive 'e yüklendi.

Bu aşamanın ardından net bağlantısıyla birlikte, Drive bağlantısı kuruldu. Burada dikkat edilmesi gereken bir diğer nokta, çalışma zamanının GPU 'ya aktivizasyonunun gerçekleştirilmesidir. Amaç Google Colab 'ın uzaktaki bilgisayarın GPU 'suna bağlantı kurması sağlanarak işlemci hacmini genişletmektir.

Ardından deneysel çalışmanın yapılacağı kitaplığın kaynağı olan Google Colaboratory'den yeni bir not defteri açıldı. Sonrasında Colab ile ortam kitaplığına bağlantı kurulumu yapıldı.

Oluşturulan veri setinin sınıflandırılması aşamasında, RELU-softmax aktivasyon fonksiyonları kullanılarak, önce RMSprop, sonrasında Adam=0.000001 ile veri sınıflandırma eğitiminin hız durumları gözlemlenerek; ayrıca bunlarla birlikte 2 farklı epoch değerlerinin girişi yapıldı; buradaki değişime göre oluşan eğitimdeki doğruluk yüzdelerini gözlemlendi.

Aşağıdaki deneysel çalışmada kullanılan kod dizinleri; sıralamaya uygun

verilerek, önemli detaylarına vurgu yapıldı.

```
from google.colab import  
drive drive.mount('/content/drive')
```

Öncelikli olarak, drive üzerinden arayüz olarak python kodlarını yazmak için kullanılacak not defterine yukarıda verilen “drive. mount”daki kod bloğu yazılıp, arayüzün drive bağlantısı için koşturulması(Run) sağlanır.

```
zip_adres = "/content/drive/MyDrive/SEM10.zip"  
! cp "{zip_adres}" .
```

Drive ‘da yer alan .zip dosyasına ulaşma adres kodunu oluşturduktan hemen sonra; python’dan komut satırına bir dizin göndermek için kopyalama komutunu “! cp” olarak oluşturulup run komutuyla işlem tamamlanır.

```
! unzip -q SEM10.zip  
! rm SEM10.zip
```

Drive ‘e yüklenen veri seti dosyasının her birinin adını yazıp; zip’ten çıkarılması için !rm komut satırı oluşturulur. Burada asıl odaklanmak istenilen nokta görsellerin ayırt edilmesinin sağlanmasıdır.

```
import os import cv2  
from tqdm import tqdm  
KATEGORILER = ["Fibres10" , "Nanowires10" , "Patterned_surface10" ,  
"Films_Coated_Surface10" , "Biological10" , "Porous_Sponge10" , "Powder10" ,  
"Tips10" , "Particles10" , "MEMS_devices_and_electrodes10"]  
DIR = "SEM10" BOYUT = 128  
veri = []
```

Kod bloğunda Operating System(os) ve Opencv(cv2) import komutuyla yüklenmesi sağlanıp (import os; import cv2); kategoriler adıyla “veri setlerinin” isimleri kategoriler kod bloğuna tek tek yazılır. Ardından verilen “DIR” komutu ile veri setinin genel adı “SEM10” olarak kaydedilir. Bundan sonraki aşamada “veri = []” boş bir array olarak tanımlanır. Kullanılan sistem CNN de görüntü sınıflandırmadır. CNN, iki boyutlu verilerde; veriye karşılık gelen filtreyi gezdirir. Çalışırken 2 kenarı eşit olan kare (görsel)lerle çalıştığı bilinmektedir.

ML’de veri boyutunun 128 olarak ayarlanmasıyla boyutun 2’nin katlarına karşılık geldiği bilinmektedir. Burada train(eğitim), validation(değer) ve test için gereken veri tahmin değerleri oluşturulur. Bu kısımda sayısal değeri veriye göre

istenilen şekilde ayarlamak mümkün olmaktadır.

```
for kategori in KATEGORILER:
```

```
    klasor_adresi = os.path.join(DIR,kategori) deger =
```

```
    KATEGORILER.index(kategori)
```

```
    for resim_adi in tqdm(os.listdir(klasor_adresi)): resim_adresi =
```

```
        os.path.join(klasor_adresi,resim_adi) resim = cv2.imread(resim_adresi,
```

```
        cv2.IMREAD_COLOR) train_size=0.8, valid_size=0.1, test_size=0.1
```

```
        if(resim is None): print(resim_adresi) print("Hata")
```

```
        else:
```

```
            resim = cv2.resize(resim,(BOYUT,BOYUT)) veri.append([resim,deger])
```

Yukarıda oluşturulan Kod betiğinde yer alan ‘for’ döngüsünün amacı betiği okumaktır. Tüm kod girişleri sağlandıktan sonra çıktı için run yapılır; burada amaç verilerin boyutlarını tekrar aynı(resize) duruma getirip; hata olabilen durumların öğrenilmesini sağlamaktır.

Aşağıda kod çıktısında değerlerin oluşumu gösterilmektedir.

```
100%|██████████| 101/101 [00:01<00:00, 82.07it/s]
```

```
SEM10/Fibres10/.DS_Store Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 82.24it/s]
```

```
SEM10/Nanowires10/.DS_Store Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 82.97it/s]
```

```
SEM10/Patterned_surface10/.DS_Store Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 78.80it/s]
```

```
SEM10/Films_Coated_Surface10/.DS_Store
```

```
Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 72.05it/s]
```

```
SEM10/Biological10/.DS_Store Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 75.90it/s]
```

```
SEM10/Porous_Sponge10/.DS_Store Hata
```

```
94%|██████████| 95/101 [00:01<00:00, 56.65it/s]
```

```
SEM10/Powder10/.DS_Store
```

```
Hata
```

```
100%|██████████| 101/101 [00:01<00:00, 69.51it/s]
```

```
100%|██████████| 101/101 [00:01<00:00, 90.95it/s]
```

SEM10/Tips10/.DS_Store Hata

100%|██████████| 101/101 [00:01<00:00, 95.15it/s]

SEM10/Particles10/.DS_Store Hata

94%|██████████| | 95/101 [00:01<00:00, 46.19it/s]

SEM10/MEMS_devices_and_electrodes10/.DS_Store

Hata

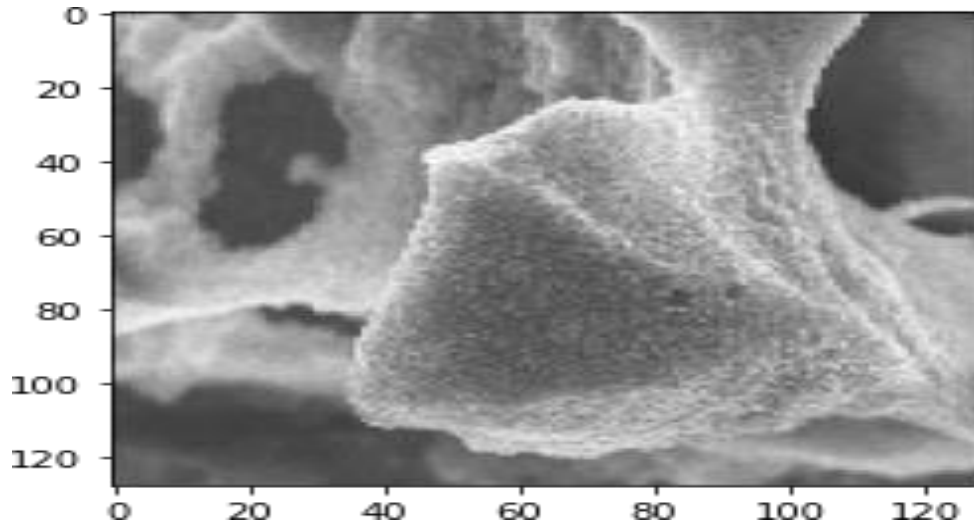
100%|██████████| 101/101 [00:02<00:00, 43.99it/s]

Çıktıda değerler incelendiğinde, gözlemlenen sonuçların yorumlanmasında, sistemin dosyaların bazılarını okuyamadığı için hata veriyor olabileceği; farklı bir düşünce olarak hataların kaynağının cv2'deki veriyi okurken ya da başlangıçta “ .jpeg ” dosyasını sıkıştırmaktan kaynaklı olabileceği şeklinde yorumlanabilir.

```
from matplotlib import pyplot
pyplot.imshow(cv2.cvtColor(veri[601][0],
cv2.COLOR_BGR2RGB))
```

Ardından verileri görselleştirmek amacıyla yukarıda ki kod betiğinde görüldüğü üzere matplotlib kütüphanesinden pyplot fonksiyonu import (içe aktarım) edilerek, oluşturulan kod betiğine bakıldığında “ veri [601][0] ” kod betiği incelendiğinde, 1. array (dizi)a kaçınıcı verinin görülmesi isteniyorsa, o değerin yazılması şeklinde düşünülebilir. Burada önemli olan bir diğer nokta sistemin aralarındaki ilişkiyi öğrenmesinin önemli olduğu; ayrıca sıralamayı öğrenmemesi, verileri ezberlememesi için önemli şartlardan biri şeklinde düşünülebilir. Elde edilen kod çıktısı aşağıdaki gibidir:

```
<matplotlib.image.AxesImage at 0x7f4d5a0ffa50>
```



Şekil 4.1 AI Tekniği ile Yapılan Sınıflandırma SEM(Toz) Görüntü Çıktısı

Kod çıktısı incelendiğinde, dikkat edilmesi gereken belki de en önemli kısım; sonuç kısmında, veri olarak o anda karıştırılan herhangi bir verinin Şekil-1’de gösterildiğidir. Nedeni eğitimde karıştırma(shuffle) yapmasından kaynaklıdır.

Devamında verilerin shuffle ‘lanması için kod betiği oluşturulur.

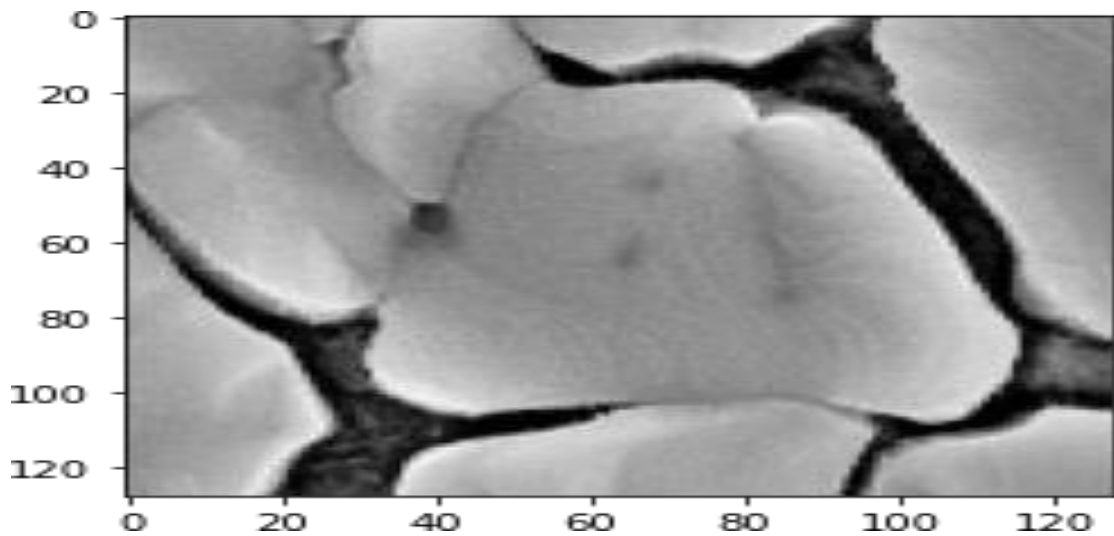
```
import random
random.shuffle(veri)
X = []
Y = []
for x,y in veri: X.append(x) Y.append(y)
del veri
```

Yukarıda ki kod betiğinde kullanılan ‘random’ kütüphanesini içe aktarmak için import edilir. ‘shuffle’komutu ile verilerin karıştırılması sağlanır. X ile ifade edilen nokta veri(data); Y ile ifade edilen noktada veri setleri (data setleri)dir. Yukarıda kod bloğu çalıştırıldığında verilerin karışıp karışmadığını anlamak için aşağıda yer alan kod bloğu oluşturulur.

```
pyplot. imshow (cv2.cvtColor(X[950],
cv2.COLOR_BGR2RGB))
print(Y[0])
```

Elde edilen çıktı:

3



Şekil 4.2 AI Tekniği İle Yapılan Sınıflandırmanın Suffle sonucu oluşan SEM(Film Kaplı Yüzey) Görüntü

Çıktısı

Verilerin yeniden boyutlandırılması için kod betiği oluşturulur.

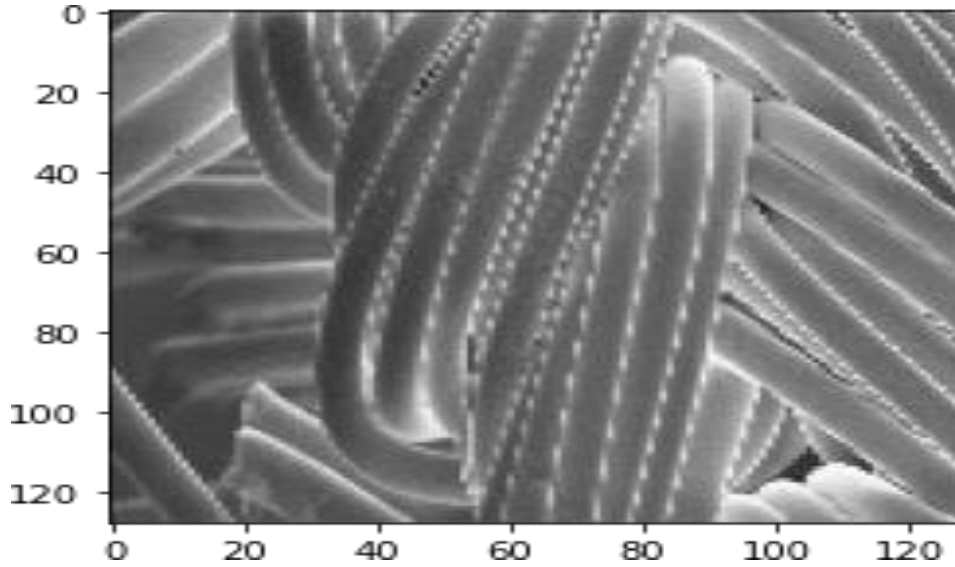
```
import numpy
X = numpy.array(X).reshape(-1,BOYUT,BOYUT,3)
Y = numpy.array(Y).reshape(-1,1) print(numpy.shape(X))
print(numpy.shape(Y))
```

Yukarıda oluşturulan kod betiğinde boyutların aynı olup olmadığını gözlemek için print(yazdır) komutu ile sonuçların gözlemlenmesi sağlanır.

```
(1000, 128, 128, 3)
(1000, 1)
```

Burada veri boyutlarının 128, 128 olduğu ve renksiz olduğu belirtilmektedir. Sonrasında görüntünün görselleştirilmesi için oluşturulan kod betiği aşağıda yer almaktadır.

```
pyplot.imshow(cv2.cvtColor(X[222], cv2.COLOR_BGR2RGB),cmap='gray')
Yukarıda 'cmap=gray' scala ile girilen kod bloğunda oluşan çıktı aşağıdadır :
<matplotlib.image.AxesImage at 0x7f4d593baa50>
```



Şekil 4.3 AI tekniği ile yapılan sınıflandırmada gray skala ile oluşan SEM(Lif) görüntü çıktısı

Yukarıda yer alan Şekil-4.3'ün renk skalası incelendiğinde görüntünün siyah-beyaz renk aralıklarında oluşturulduğu şeklinde açıklanabilir.

Devamında oluşturulan kod betiği herhangi bir sistem eğitilirken "0-255" pixel değerlerin var olduğunu göstermek içindir. Network'e input girince 0-1 arasında yer alan değer daha iyi olur.

```
X = X / 255.0
```

Yukarıdaki kodun; Normalize değerlerinin 0-1 aralığında olduğunu belirtir.

Bu sebeple yukarıda yer alan kod betiği ile verilerin normalize edilmesi sağlanır.

Deneysel çalışmanın bundan sonraki aşamasında kodlar ve çıktı sonuçları; matplotlib ile elde edilen görsel grafikleri: EK-1; EK-2; EK-3; EK-4 ile belirtilmektedir. Kodlarla yapılması planlanan durum; AI/ML alt kategorisinde DL'nin görüntü sınıflandırmada CNN tekniği kullanılarak eğitimi sağlanan SEM verilerinin içeriği oluşturduğu; 10 veri setinin; sırasıyla "ReLU-softmax" aktivasyon fonksiyonları ile önce RMSprop'ta hız ölçümü ayarlanıp; epoch sayısal değerlerinin model doğruluk yüzdelerini karşılaştırmalı görmek amacıyla; 25 ve 100 şeklinde ayarlayıp; oluşan veri sonuçlarını tablolaştırıp karşılaştırılması istenmektedir. Ardından aynı işlem "ReLU-softmax" aktivasyon fonksiyonları ile Adam = 0.000001'de hız ölçüm değer girişi sağlanıp; epoch sayısal değerlerinin model doğruluk yüzdelerini karşılaştırmalı görmek amacıyla; 25 ve 100 şeklinde ayarlayıp; oluşan veri sonuçlarını tablolaştırıp karşılaştırılması istenmektedir.

Son olarak oluşturulan 4 kategorideki en yüksek model doğruluk oranı ile diğer model doğruluk oranları kıyaslanıp; eğitim başarı yüzdesi ile sonucun CNN sınıflandırma tekniğinde yapılan eğitimde başarı sağlayıp sağlayamadığı yorumlanır.

Tablo 1 Deneysel Sonuçlar

Epoch	RMSprop	Adam = 0.000001
25	0.97	0.72
100	0.98	0.94

Yukarıda tabloda yer alan sonuçlar makine öğrenmesinin CNN'de sınıflandırma eğitiminde öncelikle aktivasyon fonksiyonu olarak RELU-softmax kullanılarak gerçekleştirildi. Eğitim hızının Adam=0.000001 değerini girerek RMSprop'a göre yavaşlatılmasından kaynaklı olabileceği düşünülen model doğruluk yüzde sonuçlarına göre yorumlanması mümkün olmaktadır. Burada öncelikli olarak RMSprop hızında; Epoch değerini 25 olarak verilmesiyle "Model doğruluk oranı", "0.97" sonucunun; Epoch değerini 100 olarak verildiğinde "Model doğruluk oranı" nda , "0.98" değerleri kıyaslandığında eğitim değerinde ki artış yüzdelerinin doğru orantılı bir şekilde arttığı yorumu yapılabilir.

Değerlerin sonucundaki farkın; eğitim sayısı arttıkça ‘‘Model doğruluk oranı’’nda ‘‘0.1’’ değerinde bir artış gösterdiği tespit edilmektedir.

Aynı şekilde Adam = 0.000001 hızında; Epoch değerini 25 olarak verilmesiyle ‘‘Model doğruluk oranı’’, ‘‘0.72’’ sonucunun; Epoch değerini 100 olarak verildiğinde ‘‘Model doğruluk oranı’’nda, ‘‘0.94’’ değerleri kıyaslandığında eğitim değerinde ki artış yüzdelik değerin doğru orantılı olarak arttığı sonucuna ulaşılabilir. Buradaki değerlerin sonucundaki farkın; eğitim sayısı arttıkça model doğruluk oranında ‘‘0,22’’ değerinde niceliksel bir artma eğilimi göstermektedir.

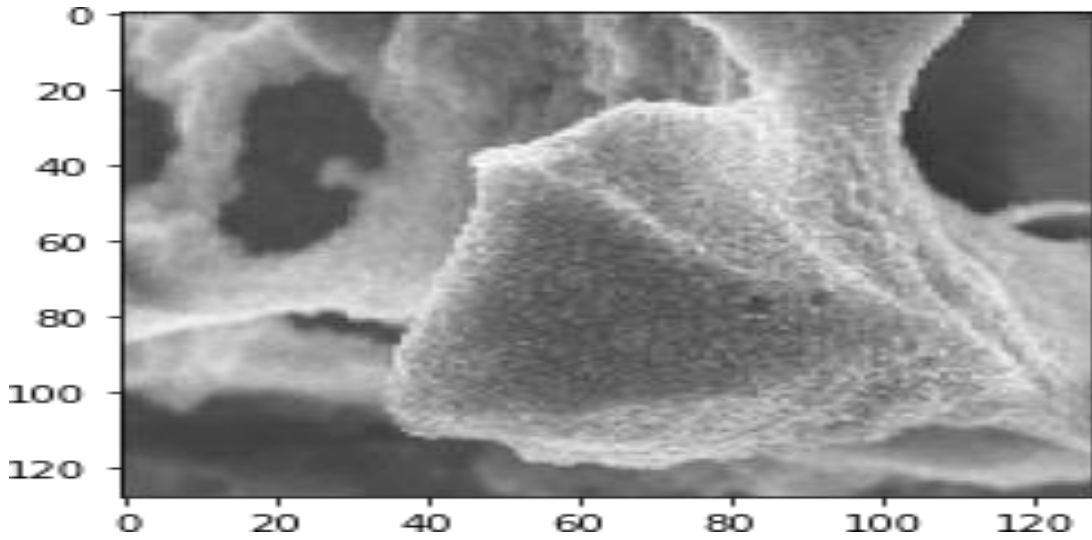
Döngü (Epoch) sayısı: Tüm verilerin eğitim sırasında ağı gösterilme sayısı olarak ifade edilip; tez kapsamında oluşturulan deney sırasında RMSprop ve Adam = 0.000001 olarak iki farklı hız döngüsünde, epoch parametreleri ‘‘25’’ ve ‘‘100’’ olarak ayarlanarak, verilerin CNN’de sınıflandırma yöntemiyle eğitimin gerçekleştirilerek; matplotlib’de elde edilen grafiksel verilerden elde edilen değer sonuçları karşılaştırılmalı olarak değerlendirilmektedir [81, 82-107].

5.TARTIŞMA SONUÇ VE ÖNERİLER

CNN’de model eğitimi gerçekleştirirken; veri setine göre farklı parametreleri işe koşmak sonuçları değerlendirmek açısından mümkün olduğunca önemlidir. Bu sebeple model eğitim fonksiyonları için giriş yapılan değerler değiştirilerek elde edilen sonuçların farklılıkları oluşturulan tablolarda açıklamalı olarak ifade edilebilir. Ayrıca görüntü eğitimde verilerin sayısının artırılması CNN’de sınıflandırma için farklı değer sonuçlarına ulaşmak için önemli farklı bir gözlem sonucuna ulaşılması açısından değerli olabilir.

Aşağıda yer alan görsel, yapay zekânın eğitiminde verilerin shuffle yapılmasının ardından oluşan çıktıdır. Burada CNN sınıflandırmada parametrelerin değiştirilmesi ile kod betiği oluşturulduğunda farklı sonuçlar verecekti. Bu değerler muhtemelen mühendislikte üretilen malzemelerin kimyasal özellikleri baz alınarak karıştırılıp kıyaslanmak istendiğinde ya da yeni bir malzeme üretimi için bu tekniklerin hangisinin uygun olabileceği araştırıldığında, bilim adamlarının koordine ve eşzamanlı çalışmaları ile ileriki çalışmalar için bir değerlendirme kriteri olabilecek yeni yazılımsal tekniklerin oluşumunda tartışma boyutunda sonuçların çıkış aşamasında önem taşımaktadır.

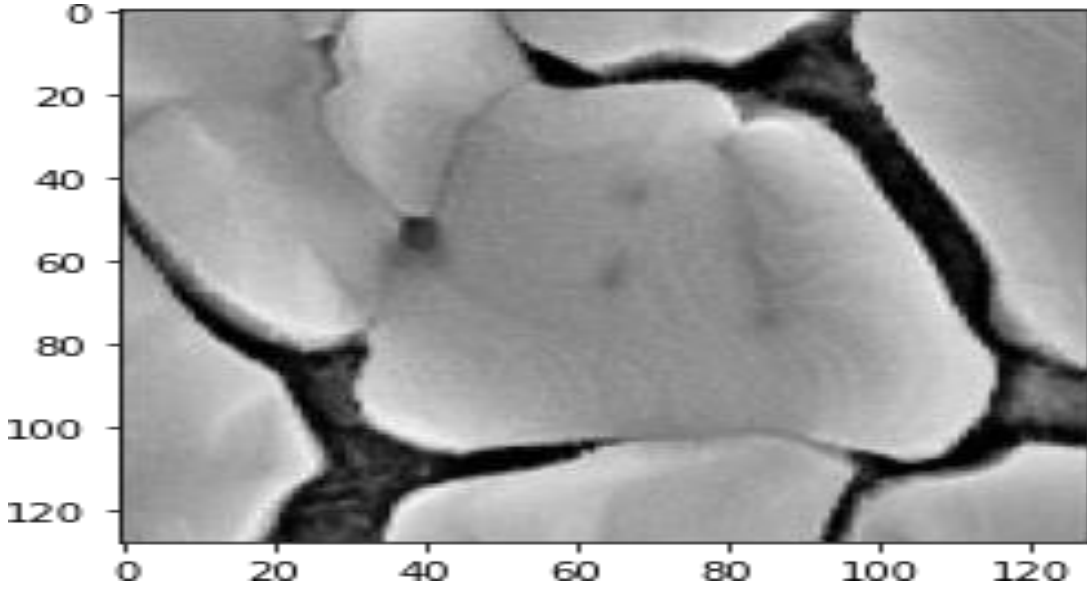
<matplotlib.image.AxesImage at 0x7f4d5a0ffa50>



Şekil 5.1 AI Tekniği ile Yapılan Sınıflandırma SEM(Toz) Görüntü Çıktısı

Deneysel çalışmada data ve data setlerinin shuffle yapılıp yapılmadığını kontrol etmek için oluşturulan kod betiğinin görsel çıktısı aşağıda ki görselde yer almaktadır.

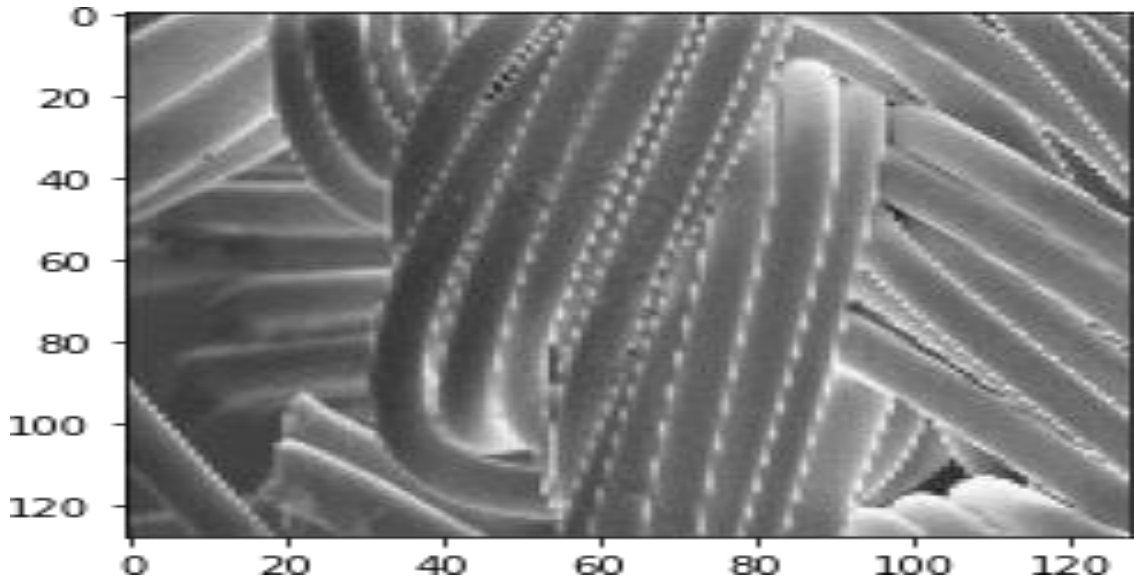
3



Şekil 5.2 AI Tekniği ile Yapılan Sınıflandırmanın Suffle sonucu oluşan SEM (Film Kaplı Yüzey) Görüntü Çıktısı

Yukarıdaki görselin veri bir önceki görselden farklı sonuç vermesi işlemin başarılı şekilde karıştırıldığını gözlemlemek için önem taşımakla birlikte, görsel verilerin farklı deneysel çalışmalar için kümeleme tekniği ile detaylandırılıp tekrarlanması, denetimli öğrenmede önemli bir gösterge olabilir.

<matplotlib.image.AxesImage at 0x7f4d593baa50>



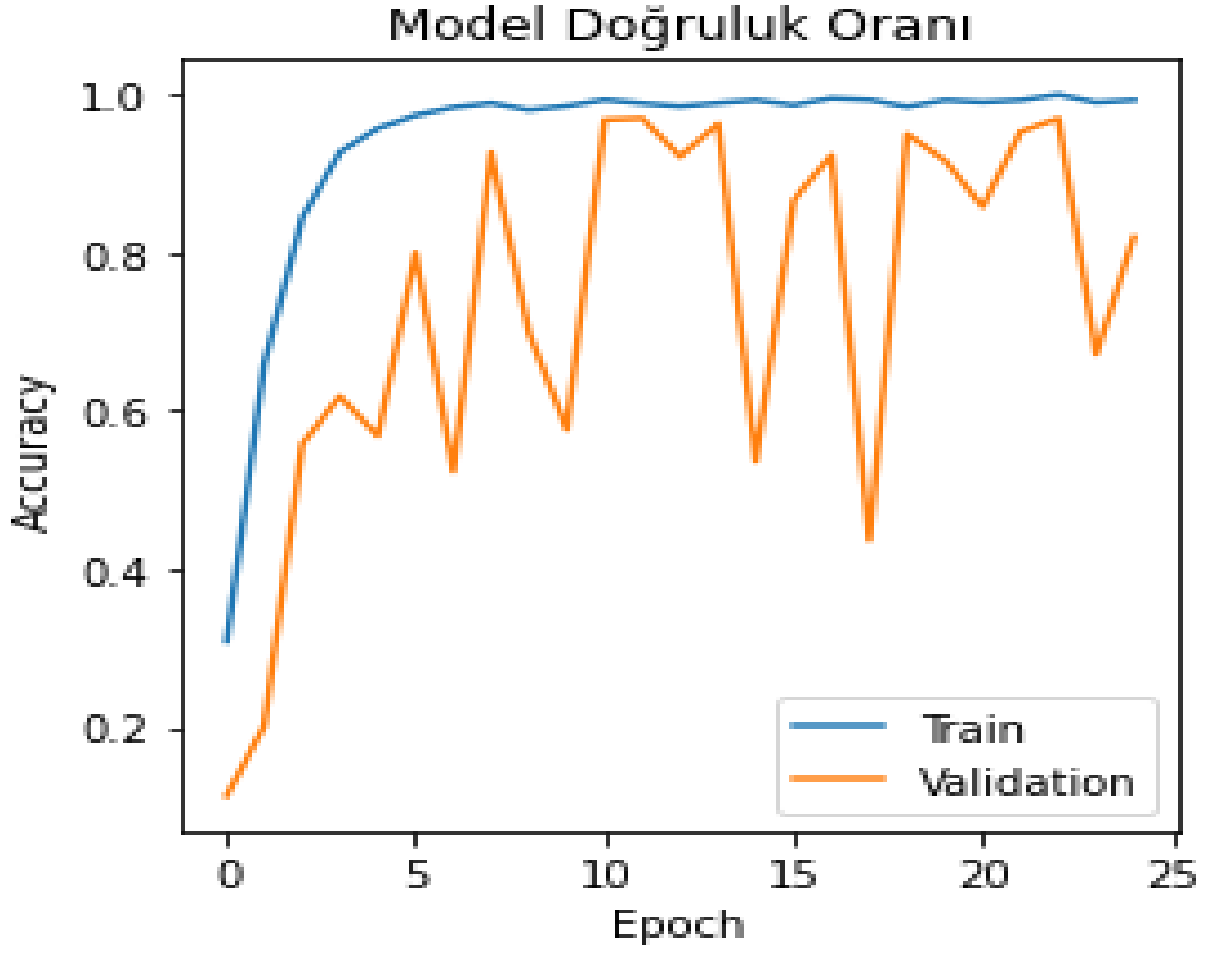
Şekil 5.3 AI tekniği ile yapılan sınıflandırmada gray skala ile oluşan SEM(Lif) görüntü çıktısı

Deneysel alıřmada kullanılan SEM grntleri genel olarak nanoboyutta ve gray skala tekniđi oluřturularak dřnldđnde, bu kod betiđi ile farklı bir renksel boyut ile karřılařma boyutunda kodların oluřumu sađlandıđında renksel boyutta deđiřikliđin oluřmadıđını yukarıdaki grsel kanıtlamaktadır.

Burada oluřan AI ile CNN eđitimde kullanılan tekniklerin grafiksel sonuları deđerleri ile ařađıda yer almaktadır:

RMSprop 25 epoch ile gerekleřen eđitim sonrasında oluřan model dođruluđu iin grafik sonu deđer ve grseli ařađıda yer almaktadır.

0.9700000286102295

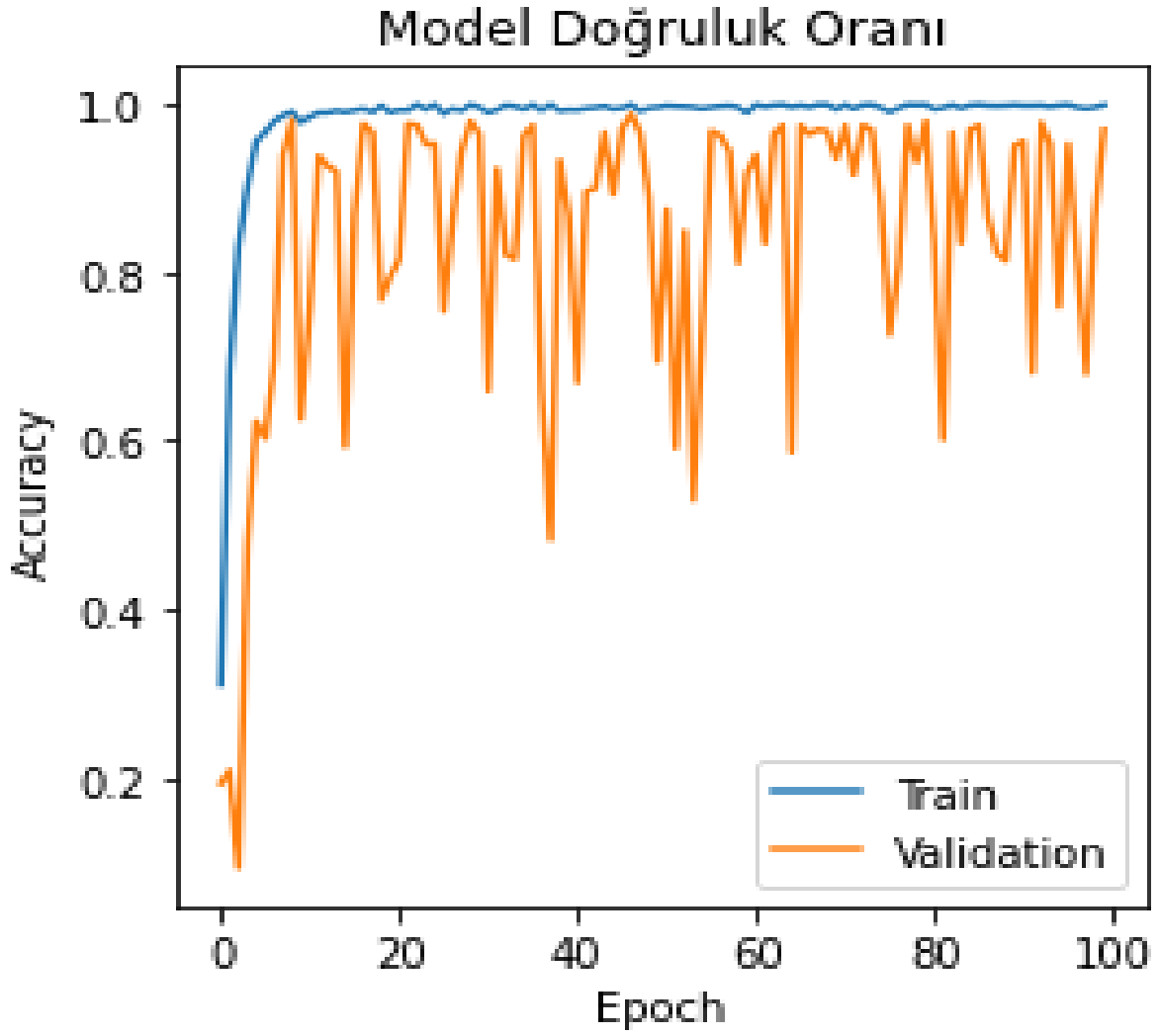


Şekil 5.4 Deneysel değerlendirilmede RMSprop fonksiyonuyla Epoch değerinin 25 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri

Grafiği yorumlamak istersek model eğitiminde değerin 1 'e yaklaşması AI ile başarılı bir eğitimin gerçekleştiğini anlatmaktadır.

RMSprop 100 epoch ile gerçekleşen eğitim sonrasında oluşan model doğruluğu için grafik sonuç değeri ve görseli aşağıda yer almaktadır.

0.9883333444595337

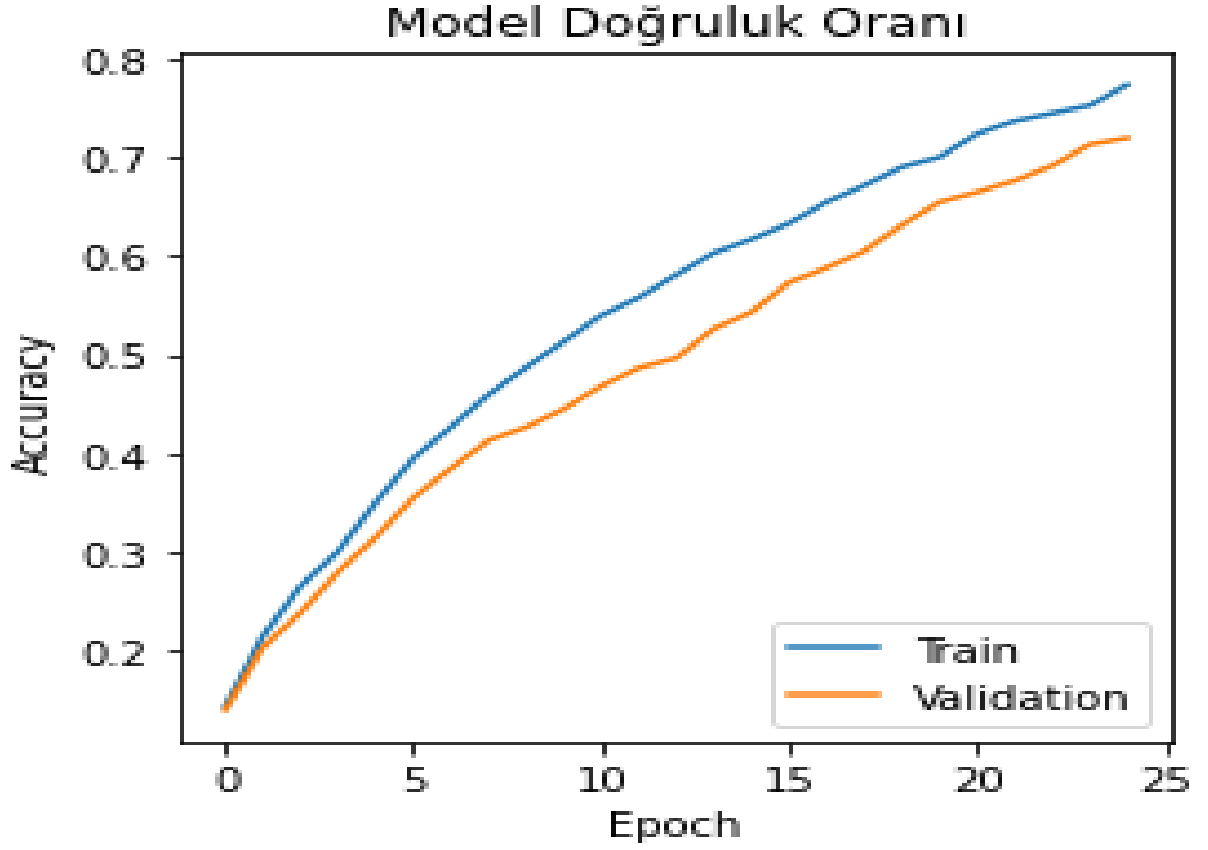


Şekil 5.5 Deneysel değerlendirilmede RMSprop fonksiyonuyla Epoch değerinin 100 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri

Grafiği yorumlamak istersek ilk eğitimden daha başarılı olmasıyla birlikte, model eğitiminde değerin 1 'e yaklaşması AI ile başarılı bir eğitimin gerçekleştiği şeklinde açıklanabilir.

Adam = 0.000001 ile 25 epochta gerçekleşen eğitim sonrasında oluşan model doğruluğu için grafik sonuç değeri ve görseli aşağıda yer almaktadır.

0.7200000286102295

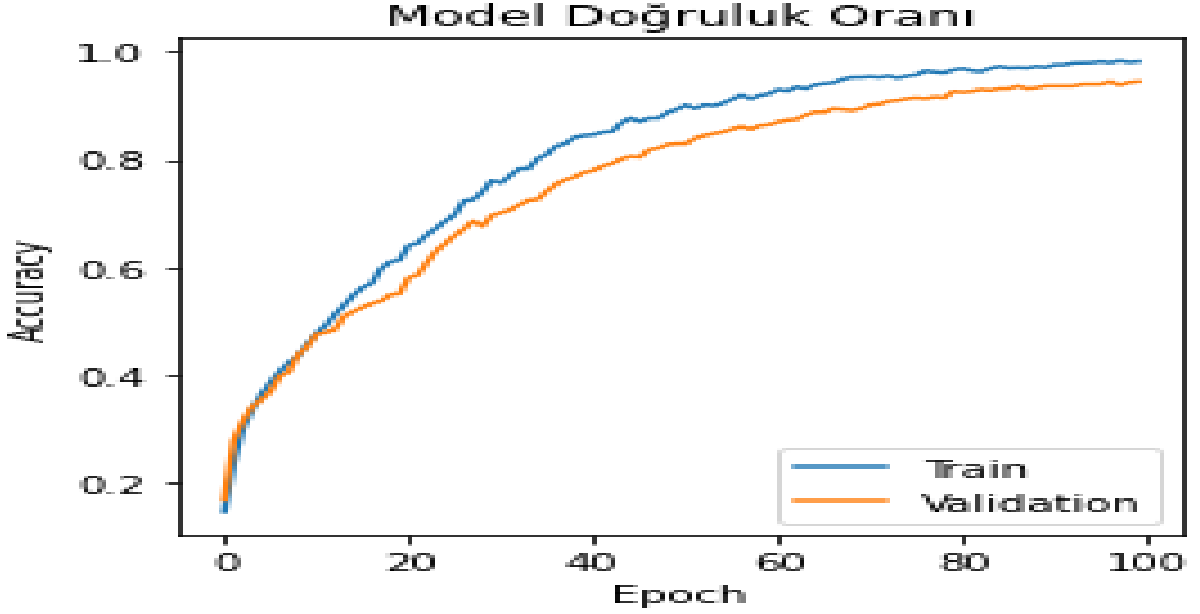


Şekil 5.6 Deneysel değerlendirilmede Adam = 0.000001 fonksiyonuyla Epoch değerinin 25 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri.

Grafiği yorumlamak istersek değer 0.72’de sonuçlandırılması RMSprop’a göre AI eğitimde düşük bir eğitim başarısı göstermesiyle birlikte korelasyon olarak bunun nedenlerinden birinin eğitilen verilerin fazla görsel ile gerçekleştirilmesi yapılacak ileriki çalışmalarda sonuçların kıyaslanması nedeniyle önemli bir boyut göstermektedir.

Adam = 0.000001 ile 100 epoch’ta gerçekleşen eğitim sonrasında oluşan model doğruluğu için grafik sonuç değeri ve görseli aşağıda yer almaktadır.

0.9449999928474426



Şekil 5.7 Deneysel değerlendirilmede Adam = 0.000001 fonksiyonuyla Epoch değerinin 100 olarak ayarlanmasıyla gerçekleşen CNN sınıflandırma eğitiminde Matplotlib ile model doğruluk oranının grafik değeri

Grafikte oluşan değer RMSprop'a göre düşük olmasının yanında Adam = 0.000001 de ve 25 epoch değeri girişi yapılarak gerçekleştirilen eğitim sonuç değerine oranla 1'e daha yakın sonuç vermesinden dolayı daha başarılı olduğu söylenebilir.

Grafiksel sonuç değerlerinin tablolaştırılarak kıyaslanması amacıyla oluşturulan tablo aşağıda yer almaktadır:

Tablo 2 Deneysel Sonuçlar

Epoch	RMSprop	Adam = 0.000001
25	0.97	0.72
100	0.98	0.94

Ayrıca elde edilen sonuç tablosundaki tüm "Model doğruluk oranları" incelendiğinde CNN ile model eğitiminde sınıflandırma değer sonucunun yüzdelik olarak en iyi değerini RELU-softmax aktivasyon fonksiyonu gerçekleşen süreçte kullanılan RMSprop ile epoch değeri olarak 100 olması bileşkesinde mümkün olduğu sonucu çıkarılabilir. Tabi ki buradaki değerler model veri setinde eğitilen verilere ve

parametrelere göre deęişiklik göstermektedir. SEM görüntülerinin tek tek incelenmesi için farklı fonksiyonlar işe koşularak daha iyi ve gerçekçi sonuçların oluşturulabileceęi, yapılacak ileriki çalışmalarla anahtar nitelięi taşımaktadır.

Bununla birlikte kullanılacak denetimli öğrenme yöntemiyle malzemelerin özellikleri kod boyutunda yazılımsal olarak geliştirilebilir. Denetimsiz öğrenme ve denetimli öğrenme yöntemleri arasında ki farklılıkların ve benzerliklerin gözlenmesi yoluyla malzemelerin; makro, mikro ve nanoboyutta ki özelliklerinde belirtilen farklılıkların nedenleri araştırılabilir. Bunun sonucunda da gerekli çözümler titiz çalışmalar sonucunda çözülebilir potansiyelde olabilir. Kimbilir bu durum bilimsel çalışmalara yeni bir çıkır açması yolunda yenilikçi bir bakış açısı oluşturabilir.

KAYNAKÇA

- [1] R. Saini, S. Saini, and S. Sharma, "Nanotechnology: the future medicine", *Journal of cutaneous and aesthetic surgery*, vol.3, no.1, pp.32–33, 2010. (<https://doi.org/10.4103/0974-2077.63301>)
- [2] A. J. Heinrich, W. D. Oliver, L. M. K. Vandersypen, et al., "Quantum-coherent nanoscience", *Nat.Nanotechnol.* no.16, pp.1318–1329, 2021. (<https://doi.org/10.1038/s41565-021-00994-1>)
- [3] T.J. Booth, M.A.B. Baker, "Chapter 32 - Nanotechnology: Building and Observing at the Nanometer Scale", *Pharmacognosy*, Academic Press, pp.633-643, 2017. (<https://doi.org/10.1016/B978-0-12-802104-0.00032-9>)
- [4] G. Caruso, L. Merlo, M. Caffo, "6 – Conclusions", Editor(s): G. Caruso, L. Merlo, M. Caffo, *Innovative Brain Tumor Therapy*, Woodhead Publishing, pp.167-173, ISBN 9781907568596, 2014. (<https://doi.org/10.1533/9781908818744.167>)
- [5] C. M. Niemeyer and C. A. Mirkin, "Nanobiotechnology: Concepts, Applications and Perspectives", Wiley-VCH, New York, 2004.
- [6] C. A. Mirkin, R. L. Letsinger, R. C. Mucic, J. J., "Storhoff, A DNA-based method for rationally assembling nanoparticles into macroscopic materials", *Nature*, v.382, pp.607–609, 1996.
- [7] E. S. Györfvay, A. O'Riordan, A. J. Quinn, G. Redmond, D. Pum, U. B. Sleytr, "Biomimetic nanostructure fabrication: non-lithographic lateral patterning and self-assembly of functional bacterial S-layers at silicon supports", *Nanolett.*, v.3, pp.315–319, 2003.
- [8] R. R. Naik, S. E. Jones, C. J. Murray, J. C. McAuliffe, R. A. Vaia, M. O. Stone, "Peptide templates for nanoparticles synthesis derived from PCR-driven phage display" *Adv. Funct. Mat.*, v.14, pp.25–30, 2004.
- [9] C. N. R. Rao and A. K. Cheetham, "Science and technology of nanomaterials: current status and future prospects" Editor: J. Mater, *Chem*, v.11, pp.2887–2894, 2001.
- [10] C.K. Schmidt, M. Medina-Sánchez, R.J. Edmondson, et al. "Engineering microrobots for targeted cancer therapies from a medical perspective", *Nat Commun*, v.11, pp.5618, 2020. (<https://doi.org/10.1038/s41467-020-19322-7>)
- [11] B.L.F. Chin, F.H. Juwono, K.S.C. Yong, "Nanotechnology and Nanomaterials for Medical Applications", Editor(s): N.M. Mubarak, S. Gopi, P. Balakrishnan, *Nanotechnology for Electronic Applications, Materials Horizons: From Nature to Nanomaterials*, Springer, Singapore, 2022. (https://doi.org/10.1007/978-981-16-6022-1_4)
- [12] G. Giri, Y. Maddahi, and K. Zareinia, "A Brief Review on Challenges in Design and Development of Nanorobots for Medical Applications", *Applied Sciences*, v.11, pp.21, 2021. (<https://doi.org/10.3390/app112110385>)
- [13] E. Oh, R. Liu, A. Nel, et al., "Meta-analysis of cellular toxicity for cadmium-containing quantum dots.", *Nature Nanotech*, v.11, pp.479–486, 2016. (<https://doi.org/10.1038/nnano.2015.33>)
- [14] A. Maleki, M. Safdari Shadloo, A. Rahmat, "Application of Artificial Neural Networks for Producing an Estimation of High-Density Polyethylene. *Polymers*

(Basel)”, v.10;12(10), pp.2319, 2020.

(doi: 10.3390/polym12102319. PMID: 33050517; PMCID: PMC7600248.)

[15] R. Shoukat, M.I. Khan, “Carbon nanotubes: a review on properties, synthesis methods and applications in micro and nanotechnology”, *Microsyst Technol* v.27, pp. 4183–4192, 2021. (<https://doi.org/10.1007/s00542-021-05211-6>)

[16] J. Devreese, “Importance of Nanosensors: Feynman's Vision and the Birth of Nanotechnology”, *MRS Bulletin*, v.32(9), pp.718-725, 2007. (<https://doi.org/10.1557/mrs2007.147>)

[17] B. Schuster, S. Weigert, D.Pum, M.Sára , U.B.Sleytr, “New method for generating tetraether lipid membranes on porous supports”, *Langmuir*, v.19, pp.2392–2397, 2003.

[18] V.N Popov, “Carbon nanotubes: properties and application”, *Mat Sci Eng.*, v.43, pp.61–102, 2004.

[19] M. Acı, M. Avcı, “Artificial neural network approach for atomic coordinate prediction of carbon nanotubes”, *Appl. Phys.* v. 122, pp.631, 2016. (<https://doi.org/10.1007/s00339-016-0153-1>)

[20] K. Akhtar, S. A. Khan, Khan S.B., A. M. Asiri, (2018) “Scanning Electron Microscopy: Principle and Applications in Nanomaterials Characterization”, In: Sharma S. (eds) *Handbook of Materials Characterization*, Springer, Cham., 2018. (https://doi.org/10.1007/978-3-319-92955-2_4)

[21] K. Akhtar, S. A. Khan, S. B. Khan, A. M. Asiri, “4. Scanning Electron Microscopy: Principle and Applications in Nanomaterials Characterization”, Springer International Publishing, *Handbook of Materials Characterization*, 2018.

[22] W. Zhou R. Apkaryan, Z. L. Wang, and D. Joy, “Fundamentals of Scanning Electron Microscopy (SEM)”, In: Zhou, W., Wang, Z.L. (eds) *Scanning Microscopy for Nanotechnology*. Springer, New York, NY. (https://doi.org/10.1007/978-0-387-39620-0_1)

[23] C. K. Schmidt, M. Medina-Sánchez,, R. J. Edmondson, et al. “Engineering microrobots for targeted cancer therapies from a medical perspective. *Nat Commun*”, v.11, pp.5618, 2020. (<https://doi.org/10.1038/s41467-020-19322-7>)

[24] R.F. Egerton, “The Scanning Electron Microscope. In: *Physical Principles of Electron Microscopy*”, Springer, Boston, MA, 2005. (https://doi.org/10.1007/0-387-26016-1_5)

[25] J. Fan, L. Fang, J. Wu, Y. Guo, Q. Dai, “From Brain Science to Artificial Intelligence”, *Engineering*, v.6, pp. 248-252 , ISSN 2095-8099, 2020. (<https://doi.org/10.1016/j.eng.2019.11.012>)

[26] S. J. P. Russell, “Norvig Artificial intelligence: a modern approach”(3rd.ed.), Pearson Education, New York (2010)

[27] M. Ashok, R. Madan, A. Joha, U. Sivarajah, “Ethical framework for Artificial Intelligence and Digital technologies”, *International Journal of Information Management*, v.62, 2022, 102433, ISSN 0268-4012, (<https://doi.org/10.1016/j.ijinfomgt.2021.102433>).

[28] A.M.Turing, “Computing Machinery and Intelligence”, In: Epstein R., Roberts G., Beber G. (eds) *Parsing the Turing Test*. Springer, Dordrecht, 2009.

(https://doi.org/10.1007/978-1-4020-6710-5_3)

[29] M. Minsky, and S. Papert, *Perceptrons*, 1969.

[30] J. McCarthy, "Defending AI research: a collection of essays and reviews", 1996.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*", 2012.

[32] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning", *Nature* v.521, pp.436–444, 2015. (<https://doi.org/10.1038/nature14539>)

[33] M. Joesch, D. Mankus, M. Yamagata, A. Shahbazi, R. Schalek, A. Suissa-Peleg, M. Meister, J.W. Lichtman, W.J. Scheirer, and J.R. Sanes, "Reconstruction of genetically identified neurons imaged by serial-section electron microscopy". *eLife*, v.5, pp.15015, 2016. (<https://doi.org/10.7554/eLife.15015>)

[34] C. Akbar, Y. Li and W. -L. Sung, "Deep Learning Algorithms for the Work Function Fluctuation of Random Nanosized Metal Grains on Gate-All-Around Silicon Nanowire MOSFETs," in *IEEE Access*, vol. 9, pp. 73467-73481, 2021. (doi: 10.1109/ACCESS.2021.3079981)

[35] D.M. Dimiduk, E.A. Holm, and S.R. Niezgod, "Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering", *Integr Mater Manuf Innov*, v.7, pp.157–172 2018. (<https://doi.org/10.1007/s40192-018-0117-8>)

[36] M.I. Jordan, T.M. Mitchell, "Machine learning: trends, perspectives, and prospects" *Science*, v.349 (6245), pp.255–260, 2015.

[37] A. M. Saxe, J. L. McClelland, S. Ganguli, "A mathematical theory of semantic development in deep neural networks", *Proceedings of the National Academy of Sciences*, v.116 (23), pp.11537-11546, 2019. (<https://doi.org/10.1073/pnas.1820226116>)

[38] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks", *Proceedings of the national academy of sciences*, v.114(13), pp.3521-3526, 2017. (<https://www.pnas.org/doi/10.1073/pnas.1611835114>)

[39] J. Ma, R.P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships" *J. Chem. Inf.* v.55, pp.263–274 2015.

[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks" In *Proc. 14th International Conference on Artificial Intelligence and Statistics*, pp.315–323, 2011.

[41] D. Jerez, E. Stuart, K. Schmitt, et al., "A deep learning approach to identifying immunogold particles in electron microscopy images". *Sci Rep* v.11, pp.7771, 2021. (<https://doi.org/10.1038/s41598-021-87015-2>)

[42] A. Agrawal, A. Choudhary, "Deep materials informatics: Applications of deep learning in materials science", *MRS Communications* v.9, pp.779–792, 2019. (<https://doi.org/10.1557/mrc.2019.73>)

[43] A. Jakupović, Ž. Kovačević, L. Gurbeta and A. Badnjević, "Review of artificial neural network application in nanotechnology," 2018 7th Mediterranean Conference on

Embedded Computing (MECO), pp.1-4, 2018. (doi: 10.1109/MECO.2018.8406006)

[44] D. H. Kim, T. I. Zohdi, "Tool path optimization of selective laser sintering processes using deep learning", *Comput Mech* v.69, pp.383–401, 2022. (<https://doi.org/10.1007/s00466-021-02079-1>)

[45] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning", MIT Press, Cambridge, 2016.

[46] K. He et al., "Deep residual learning for image recognition", In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, ABD, 2016, pp 770–778.

[47] Y. LeCun, et al., "Handwritten digit recognition with a back-propagation network", In Proc. Advances in Neural Information Processing Systems pp.396–404, 1990.

[48] R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, R. Asahi,"Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics",*Acta Materialia*,v. 141, pp.29-38,ISSN 1359-6454,2017. (<https://doi.org/10.1016/j.actamat.2017.09.004>)

[49] B.T. Boukaye, B. Kamsu-Foguem, F. Tangara,"Deep convolution neural network for image recognition",*Ecological Informatics*,v.48,pp.257-268,ISSN 1574-9541, 2018. (<https://doi.org/10.1016/j.ecoinf.2018.10.002>)

[50] B.P.Sowmya, M.C.Supriya, "Convolutional Neural Network (CNN) Fundamental Operational Survey", In: Favorskaya, M.N., Peng, SL., Simic, M., Alhadidi, B., Pal, S. (eds) *Intelligent Computing Paradigm and Cutting-edge Technologies. ICICCT 2020. Learning and Analytics in Intelligent Systems*, v.21, Springer, Cham, 2021. (https://doi.org/10.1007/978-3-030-65407-8_21)

[51] J. Wei, X.Chu, X-Y. Sun, et al.,"Machine learning in materials science", *InfoMat.*, v.1, pp.338– 358,2019. (<https://doi.org/10.1002/inf2.12028>)

[52] Y. Tang,R. Salakhutdinov, and G. Hinton,"Tensor Analyzers", Proceedings of the 30th International Conference on Machine Learning, in Proceedings of Machine Learning Research", v.28(3), pp.63-171, 2013. (<https://proceedings.mlr.press/v28/tang13.html>)

[53] H. Wang, Y. Xin, "Wavelet-Based Contourlet Transform and Kurtosis Map for Infrared Small Target Detection in Complex Background. *Sensors (Basel)*", v.3, pp.755, 2020. (doi: 10.3390/s20030755)

[54] Y. Zhang, C. Gao, S. Pan, Z. Li, Y. Xu, and H. Qiu, "A Score-Level Fusion of Fingerprint Matching with Fingerprint Liveness Detection", *IEEE Access*, v. 8, s.183391-183400, 2020. (doi: 10.1109/ACCESS.2020.3027846)

[55] L. Deng, G. Hinton and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8599-8603, 2013. (doi: 10.1109/ICASSP.2013.6639344)

[56] C. E. Hachimi, S. Belaqziz, S. Khabba, A. Chehbouni, "Data Science Toolkit: An all-in-one python library to help researchers and practitioners in implementing data science-related algorithms with less effort",*Software Impacts*,v.12,2022.

- [57] P. Virtanen, R. Gommers, T. E. Oliphant, et al., “SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*” v.17, pp.261–272, 2020. (<https://doi.org/10.1038/s41592-019-0686-2>)
- [58] B. Mahltig, and T. Grethe, “High-Performance and Functional Fiber Materials—A Review of Properties, Scanning Electron Microscopy SEM and Electron Dispersive Spectroscopy EDS,” *Textiles*, vol. 2, no. 2, pp. 209–251, Apr. 2022, doi: 10.3390/textiles2020012.
- [59] W. Pelton, “Distinguishing the Cause of Textile Fiber Damage Using the Scanning Electron Microscope (SEM).” *Journal of Forensic Sciences* 40, pp.874-882, 1995.
- [60] J. Jeevanandam, A. Sundaramurthy, V. Sharma, C. Murugan, K. Pal, M. Hamada A. Kodous, M. K. Danquah, “Chapter 4 - Sustainability of One-Dimensional Nanostructures: Fabrication and Industrial Applications”, Editor(s): Gyorgy Szekely, Andrew Livingston, Sustainable Nanoscale Engineering, Elsevier, pp.83-113 2020, ISBN 9780128146811, <https://doi.org/10.1016/B978-0-12-814681-1.00004-7>.
- [61] A. Berardo, G. Costagliola, S. Ghio, M. Boscardin, F. Bosia, and N. Pugno, “An experimental-numerical study of the adhesive static and dynamic friction of micro-patterned soft polymer surfaces”, *Materials & Design*, pp.181, 2019.
- [62] M. Niwa, Y. Hiraishi, “Quantitative analysis of visible surface defect risk in tablets during film coating using terahertz pulsed imaging” *International Journal of Pharmaceutics*, v.461, pp.342-350, ISSN 0378-5173, 2014, <https://doi.org/10.1016/j.ijpharm.2013.11.051>.
- [63] <https://link.springer.com/book/10.1007/978-3-642-80834-0#affiliations>
- [64] A. Bucciarelli, T. Muthukumar, J. S. Kim, W. K. Kim, A. Quaranta, D. Maniglio, G. Khang, and A. Motta, “Preparation and Statistical Characterization of Tunable Porous Sponge Scaffolds using UV Cross-linking of Methacrylate-Modified Silk Fibroin” *ACS Biomaterials Science & Engineering*, v.5(12), pp.6374-6388, 2019, DOI: 10.1021/acsbomaterials.9b00814.
- [65] R. K. Ramakrishnan, V. V. T. Padil, M. Škodová, S. Waclawek, M. Černík, and S. Agarwal, “Hierarchically Porous Bio-Based Sustainable Conjugate Sponge for Highly Selective Oil/Organic Solvent Absorption”, *Adv. Funct. Mater.*, pp.31. 2021. (<https://doi.org/10.1002/adfm.202100640>)
- [66] M. Albu, S. Mitsche, M. Nachtnebel, et al., “Microstructure Investigations of Powders and Additive Manufactured Parts”, *Berg Huettenmaenn Monatsh*, v.165, pp.169–174, 2020. <https://doi.org/10.1007/s00501-020-00958-3>.
- [67] C. Shi, D. Luu, Q. Yang, et al., “Recent advances in nanorobotic manipulation inside scanning electron microscopes”, *Microsyst Nanoeng*, v.2, pp.16024, 2016. (<https://doi.org/10.1038/micronano.2016.24>)
- [68] B. Rühle, J. F. Krumrey, and V. D. Hodoroaba, “Workflow towards automated segmentation of agglomerated, non-spherical particles from electron microscopy images using artificial neural networks”. *Sci Rep*, v.11, pp.4942, 2021. (<https://doi.org/10.1038/s41598-021-84287-6>)
- [69] <https://www.upstate.edu/pathenvi/>

- [70] <https://www.nanowerk.com/mems-explained.php>
- [71] M. H. Modarres, R. Aversa, S. Cozzini, et al., “Neural Network for Nanoscience Scanning Electron Microscope Image Recognition”, *Sci Rep*, v.7, pp.13282,2017. (<https://doi.org/10.1038/s41598-017-13565-z>)
- [72] J. Raitoharju, “Chapter 3 - Convolutional neural networks”, Editor(s): A.I. Iosifidis, A. Tefas, *Deep Learning for Robot Perception and Cognition*, Academic Press, pp. 35-69, ISBN 9780323857871, 2022, <https://doi.org/10.1016/B978-0-32-385787-1.00008-7>.
- [73] K.C. Santosh, N. Das, S. Ghosh, “Chapter 2 - Deep learning: a review”, Editor(s): K.C. Santosh, N. Das, S. Ghosh, *In Primers in Biomedical Imaging Devices and Systems, Deep Learning Models for Medical Imaging*, Academic Press, pp. 29-63, ISBN 9780128235041, 2022, <https://doi.org/10.1016/B978-0-12-823504-1.00012-X>.
- [74] A. Zaras, Nikolaos Passalis, A. Tefas, “Chapter 2 - Neural networks and backpropagation”, Editor(s): A. Iosifidis, A. Tefas, *Deep Learning for Robot Perception and Cognition*, Academic Press, pp. 17-34, ISBN 9780323857871, 2022, <https://doi.org/10.1016/B978-0-32-385787-1.00007-5>.
- [75] Q. Zhang, J. Yang, X. Zhang, T. Cao, “SO-softmax loss for discriminable embedding learning in CNNs”, *Pattern Recognition*, v.131, pp.108877, ISSN 0031-3203, 2022, <https://doi.org/10.1016/j.patcog.2022.108877>.
- [76] Y. Wei, Q. Duan, R. Yuan, X. Yan, Y. Yang, “Dropout neuronal unit with tunable probability based on NbOx stochastic memristor for efficient suppression of overfitting”, *Microelectronic Engineering*, v. 259, pp.111778, ISSN 0167-9317, 2022, <https://doi.org/10.1016/j.mee.2022.111778>.
- [77] S. Saxena, S. Tripathi, T.S.B. Sudarshan, “An intelligent facial expression recognition system with emotion intensity classification”, *Cognitive Systems Research*, v.74, pp. 39-52, ISSN 1389-0417, 2022, <https://doi.org/10.1016/j.cogsys.2022.04.001>.
- [78] Y. Zheng, B. Kenji Iwana, M. Imran Malik, S. Ahmed, W. Ohyama, S. Uchida, “Learning the micro deformations by max-pooling for offline signature verification”, *Pattern Recognition*, v.118, 108008, ISSN 0031-3203, 2021, <https://doi.org/10.1016/j.patcog.2021.108008>.
- [79] J. Jin, A. Dundar, and E. Culurciello, “Flattened Convolutional Neural Networks for Feedforward Acceleration”, 2014, <https://www.arxiv-vanity.com/papers/1412.5474/>.
- [80] H.A. Elmarakeby., J. Hwang, R. Arafah, et al., “Biologically informed deep neural network for prostate cancer discovery”, *Nature* v.598, pp.348–352, 2021. <https://doi.org/10.1038/s41586-021-03922-4>
- [81] Y. Zheng, et al. “Transfer of Learning in the Convolutional Neural Networks on Classifying Geometric Shapes Based on Local or Global Invariants.” *Frontiers in computational neuroscience*, v.15, pp.63714, 19 Feb. 2021, doi:10.3389/fncom.2021.637144
- [82] https://ars.els-cdn.com/content/image/1-s2.0-S0039914018302054-fx1_lrg.jpg
- [83] <https://www.alamy.com/stock-photo/nano-molecule.html?sortBy=relevant>
- [84] <https://nanoteknoloji.org/tag/nanotip/>

- [85] <https://nano.aku.edu.tr/2016/05/12/nano-3/>
- [86] <https://ahmetakgonul.weebly.com/kullan305m-alanlar305.html>
- [87] <https://www.wired.co.uk/article/ibm-carbon-nanotube-breakthrough-processors-silicon>
- [88] <https://www.britannica.com/technology/scanning-electron-microscope>
- [89] <https://www.aitimejournal.com/@nisha.arya.ahmed/what-is-artificial-intelligence-ai>
- [90] K. K. Htike and O. O. Khalifa, "Comparison of supervised and unsupervised learning classifiers for human posture recognition," *International Conference on Computer and Communication Engineering (ICCCE'10)*, 2010, pp. 1-6, doi: 10.1109/ICCCE.2010.5556749.
- [91] <https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b>
- [92] https://www.researchgate.net/figure/Difference-between-Machine-Learning-and-Deep-Learning_fig4_338585724/download
- [93] Yamashita, R., Nishio, M., Do, R.K.G. *et al.*. "Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9", pp.611–629,2018.
(<https://doi.org/10.1007/s13244-018-0639-9>)
- [94] İ. Topcu, H.O. Gülsoy, A.N. Güllüoğlu, "Experimental investigation on mechanical properties of Multi Wall Carbon Nanotubes (MWCNT) reinforced aluminium metal matrix composites", *Gazi University Journal of Science*, v.32 (1), pp.286-298, 2019.
- [95] A. Rauniyar, D. Hagos, U. Bağcı, "Federated Learning for Medical Applications: A Taxonomy, Current Trends, Challenges, and Future Research Directions", 2022.
- [96] J. N. Stemmer, H. Celik, E. Krupinski, *et al.* Eye Tracking for Deep Learning Segmentation Using Convolutional Neural Networks. *J Digit Imaging* 32, 597–604,2019.(<https://doi.org/10.1007/s10278-019-00220-4>)
- [97] İ. Topcu, H.O. Gülsoy, A.N. Güllüoğlu, "Çok Duvarlı CNT Partikül Takviyeli Ti6Al4V Alaşım Esaslı Kompozitlerin Statik Yükler Altında Malzemelerin Sürtünme Davranışının Değerlendirilmesi", vol.32, pp.286-298, 2019.
- [98] S. Hussein, P. Kandel, C. W. Bolan, M. B. Wallace and U. Bagci, et al, "Lung and Pancreatic Tumor Characterization in the Deep Learning Era: Novel Supervised and Unsupervised Learning Approaches," in *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1777-1787, Aug. 2019, doi: 10.1109/TMI.2019.2894349.
- [99] A.I. Oludare, J. Aman, E. O., Abiodun, Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, Humaira Arshad, "State-of-the-art in artificial neural network applications: A survey, *Heliyon*", vol. 4, n. 11,2018.
- [100] McCarthy J., "What is artificial intelligence", Computer Science Department, Stanford University. Available from: <http://www-formal.stanford.edu/jmc/whatisai.pdf>
- [101] <https://www.javatpoint.com/application-of-ai>
- [102] https://serc.carleton.edu/research_education/geochemsheets/techniques/SEM.html
- [103] <https://pubmed.ncbi.nlm.nih.gov/30519653/>

[104] CS231n: Görsel Tanıma için Evrişimli Sinir Ağları — Convolutional Neural Networks for Visual Recognition

[105] <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9>

[106] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3682481

[107] <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9>

EKLER

EK-1

RELU-softamax RMSprop hızı ile ve Epoch = 25 değeri verilerek gerçekleştirilen kod dizini:

```
from tensorflow.keras import optimizers from tensorflow.keras import Sequential from
tensorflow
import one_hot
from tensorflow.keras.callbacks import ModelCheckpoint from tensorflow.keras.layers
import Conv2D, Activation,
BatchNormalization, Flatten, Dense, MaxPool2D
model = Sequential()
model.add(Conv2D(filters=128,kernel_size=(3,3),input_shape=X[0].shape,ac
tivation='relu'))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(BatchNormalization()) model.add(Flatten())
model.add(Dense(64,activation='relu')) model.add(Dense(32,activation='relu'))
model.add(Dense(10,activation='softmax'))
print(numpy.shape(Y)) model.compile(loss='sparse_categorical_crossentropy',
optimizer=optimizers.RMSprop(), metrics=['acc'])
mc = ModelCheckpoint('en_ iyi.h5',save_best_only=True,monitor='val_loss',mode=
'min')
model.fit(X,Y,batch_size=32,epochs=25,shuffle=True,validation_split=0.2,
callbacks=[mc])
```

Ek-1 Sonuç:

(1000, 1)

Epoch 1/25

75/75 [=====] - 4s 48ms/step - loss: 3.2092 - acc:
0.3083 - val_loss:

2.3164 - val_acc: 0.1133 Epoch 2/25

75/75 [=====] - 3s 44ms/step - loss: 1.0513 - acc:
0.6596 - val_loss:

2.2984 - val_acc: 0.2017 Epoch 3/25

75/75 [=====] - 3s 43ms/step - loss: 0.5227 - acc:
0.8446 - val_loss:

1.4406 - val_acc: 0.5583 Epoch 4/25

75/75 [=====] - 3s 43ms/step - loss: 0.2473 - acc:
0.9271 - val_loss:

1.2262 - val_acc: 0.6183 Epoch 5/25

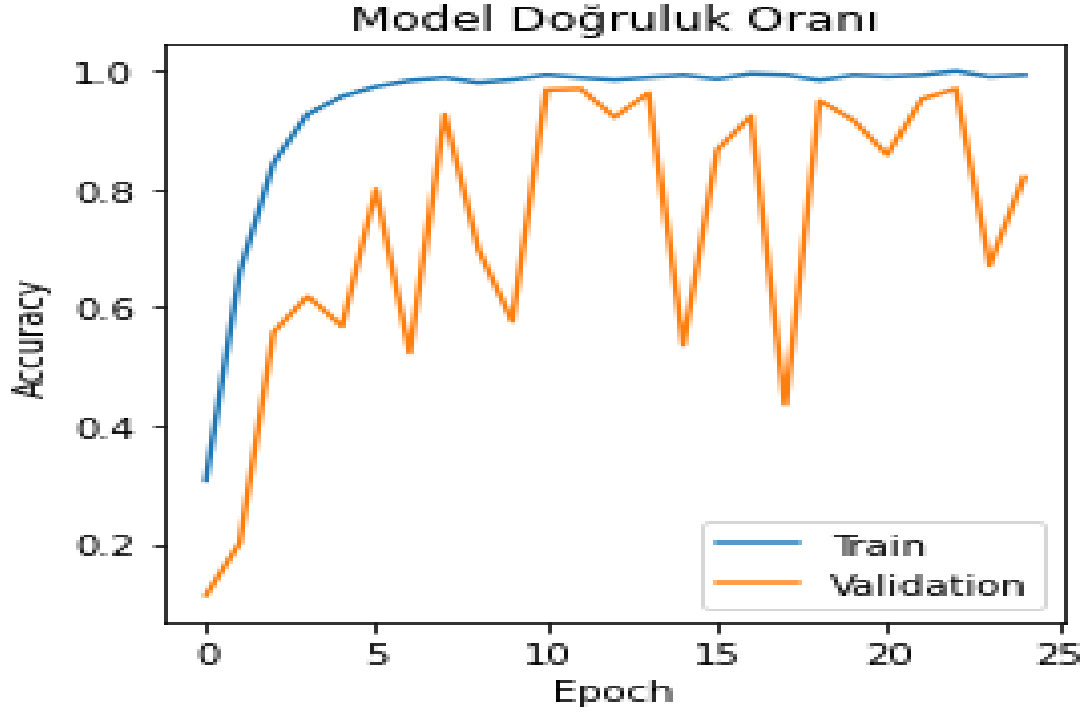
75/75 [=====] - 3s 42ms/step - loss: 0.1641 - acc:
0.9575 - val_loss:
1.5183 - val_acc: 0.5683 Epoch 6/25
75/75 [=====] - 3s 43ms/step - loss: 0.0996 - acc:
0.9737 - val_loss:
0.8282 - val_acc: 0.8000 Epoch 7/25
75/75 [=====] - 3s 42ms/step - loss: 0.0518 - acc:
0.9837 - val_loss:
2.5564 - val_acc: 0.5233 Epoch 8/25
75/75 [=====] - 3s 44ms/step - loss: 0.0370 - acc:
0.9892 - val_loss:
0.4689 - val_acc: 0.9267 Epoch 9/25
75/75 [=====] - 3s 45ms/step - loss: 0.0893 - acc:
0.9804 - val_loss:
2.1718 - val_acc: 0.6983 Epoch 10/25
75/75 [=====] - 3s 42ms/step - loss: 0.0825 - acc:
0.9854 - val_loss:
4.7294 - val_acc: 0.5767 Epoch 11/25
75/75 [=====] - 3s 44ms/step - loss: 0.0253 - acc:
0.9933 - val_loss:
0.3745 - val_acc: 0.9683 Epoch 12/25
75/75 [=====] - 3s 43ms/step - loss: 0.0459 - acc:
0.9887 - val_loss:
0.3830 - val_acc: 0.9700 Epoch 13/25
75/75 [=====] - 3s 43ms/step - loss: 0.0543 - acc:
0.9850 - val_loss:
0.4821 - val_acc: 0.9217 Epoch 14/25
75/75 [=====] - 3s 43ms/step - loss: 0.0568 - acc:
0.9892 - val_loss:
0.4591 - val_acc: 0.9633 Epoch 15/25
75/75 [=====] - 3s 43ms/step - loss: 0.0322 - acc:
0.9929 - val_loss:
4.8130 - val_acc: 0.5367 Epoch 16/25
75/75 [=====] - 3s 43ms/step - loss: 0.0452 - acc:
0.9867 - val_loss:
1.2709 - val_acc: 0.8667 Epoch 17/25
75/75 [=====] - 3s 43ms/step - loss: 0.0212 - acc:
0.9954 - val_loss:
0.7177 - val_acc: 0.9233 Epoch 18/25
75/75 [=====] - 3s 43ms/step - loss: 0.0242 - acc:
0.9933 - val_loss:
10.7047 - val_acc: 0.4367 Epoch 19/25
75/75 [=====] - 3s 42ms/step - loss: 0.0814 - acc:

```
0.9846 - val_loss:
0.7302 - val_acc: 0.9500 Epoch 20/25
75/75 [=====] - 3s 43ms/step - loss: 0.0438 - acc:
0.9929 - val_loss:
0.9474 - val_acc: 0.9167 Epoch 21/25
75/75 [=====] - 3s 42ms/step - loss: 0.0375 - acc:
0.9904 - val_loss:
1.1095 - val_acc: 0.8583 Epoch 22/25
75/75 [=====] - 3s 42ms/step - loss: 0.0324 - acc:
0.9929 - val_loss:
0.7004 - val_acc: 0.9533 Epoch 23/25
75/75 [=====] - 3s 43ms/step - loss: 1.3464e-04 -
acc: 1.0000 - val_loss:
0.6843 - val_acc: 0.9700 Epoch 24/25
75/75 [=====] - 3s 42ms/step - loss: 0.0340 - acc:
0.9900 - val_loss:
3.1324 - val_acc: 0.6717 Epoch 25/25
75/75 [=====] - 3s 42ms/step - loss: 0.0301 - acc:
0.9929 - val_loss:
1.7651 - val_acc: 0.8200
<keras.callbacks.History at 0x7f4d52fc3a10>
```

Ek-1 için verilen görselleştirilmesi için ‘‘ matplotlib ‘‘ oluşturulan kod sistemi:

```
from matplotlib import pyplot
pyplot.figure(figsize=(4,4)) pyplot.plot(model.history.history['acc'])
pyplot.plot(model.history.history['val_acc']) pyplot.title('Model Doğruluk Oranı')
pyplot.ylabel('Accuracy') pyplot.xlabel('Epoch')
pyplot.legend(['Train', 'Validation'], loc='lower right') pyplot.savefig('200.png',dpi=200)
print(max(model.history.history['val_acc']))
```

Grafik sonuç değeri ve görseli Ek.1:
0.9700000286102295



Şekil Ek-1.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi

```
model.save('model.h5')  
from tensorflow.keras.models import load_model yuklenen_model =  
load_model('model.h5')  
from sklearn.metrics import confusion_matrix, classification_report  
print(numpy.shape(Y))  
tahminler = yuklenen_model.predict(X) tahminler = numpy.argmax(tahminler,axis=1)  
print(numpy.shape(tahminler))  
print(confusion_matrix(Y,tahminler)) print(classification_report(Y,tahminler))
```

Ek-1 için oluşan yukarıda ki kod betiğinin tablo çıktısı:

(1000, 1)

(1000,)

```
[[294  0  0  0  0  0  3  3  0 0]
 [ 0 285  6  0  0  0  3  6  0 0]
 [ 0  0 294  0  0  0  0  3  3 0]
 [ 0  0  3 279  0  0  0 18  0 0]
 [15  0 75 18 90  0  0 102  0 0]
 [24  0 15 30  0 207  6 18  0 0]
 [ 0  0  0  6  0  0 291  3  0 0]
 [ 0  0  0  0  0  0  0 300  0 0]
 [ 6  0  6  0  0  0  0  33 255 0]
 [12  0  0  0  0  0 12 15  0 261]]
```

	precision	recall	f1-score	support
0	0.84	0.98	0.90	100
1	1.00	0.95	0.97	100
2	0.74	0.98	0.84	100
3	0.84	0.93	0.88	100
4	1.00	0.30	0.46	100
5	1.00	0.69	0.82	100
6	0.92	0.97	0.95	100
7	0.60	1.00	0.75	100
8	0.99	0.85	0.91	100
9	1.00	0.87	0.93	100
accuracy			0.85	1000
macro avg	0.89	0.85	0.84	1000
weighted avg	0.89	0.85	0.84	1000

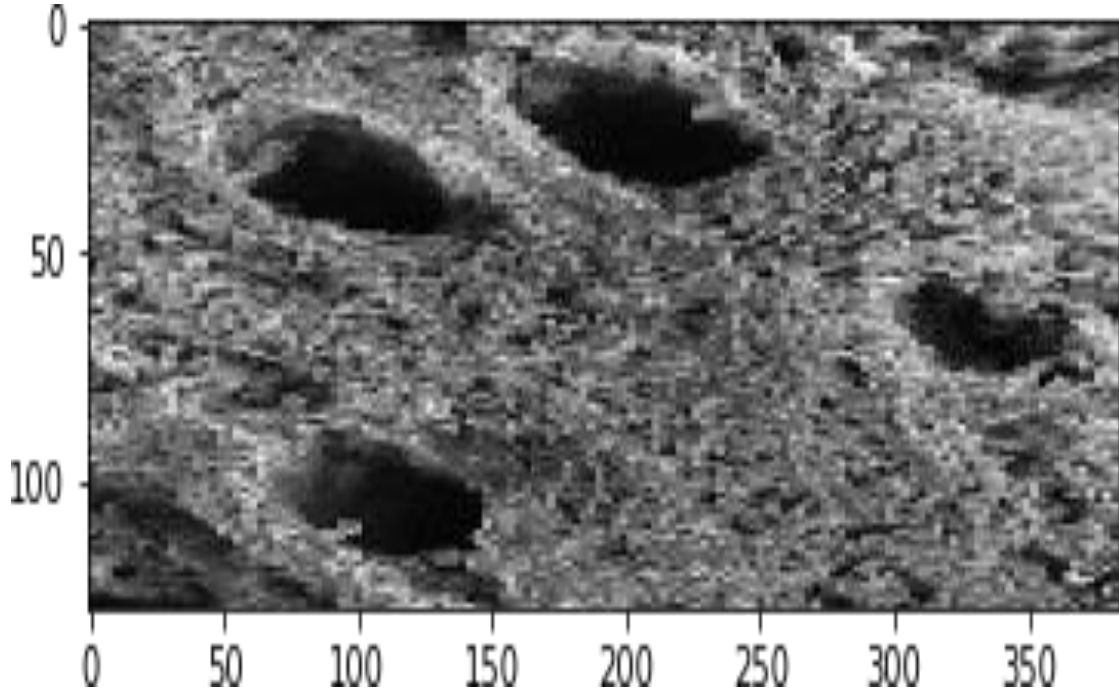
Eğitim sonucunda oluşan şekil için kod betiği :

```
print(tahminler[0]) pyplot.imshow(numpy.column_stack(X[100]),cmap='gray')
```

Eđitim sonucunda, tahmini olarak oluřan Őekil Ek.1:

3

<matplotlib.image.AxesImage at 0x7f4d4ba3dd90>



Őekil Ek-1.2: AI tekniđi ile yapılan sınıflandırmada oluřan SEM(Lif) görüntü çıktıısı

EK-2

RELU-softamax RMSprop hızı ile ve Epoch = 100 değeri verilerek gerçekleştirilen kod dizini:

```
from tensorflow.keras import optimizers from tensorflow.keras import Sequential from tensorflow import one_hot
from tensorflow.keras.callbacks import ModelCheckpoint from tensorflow.keras.layers import Conv2D, Activation,
BatchNormalization, Flatten, Dense, MaxPool2D
model = Sequential()
model.add(Conv2D(filters=128,kernel_size=(3,3),input_shape=X[0].shape,activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(BatchNormalization()) model.add(Flatten())
model.add(Dense(64,activation='relu')) model.add(Dense(32,activation='relu'))
model.add(Dense(10,activation='softmax'))
print(numpy.shape(Y)) model.compile(loss='sparse_categorical_crossentropy',
optimizer=optimizers.RMSprop(), metrics=['acc'])
mc = ModelCheckpoint('en_ iyi.h5',save_best_only=True,monitor='val_loss',mode='min')
model.fit(X,Y,batch_size=32,epochs=100,shuffle=True,validation_split=0.2
,callbacks=[mc])
```

Ek-2 Sonuç:

(1000, 1)

Epoch 1/100

75/75 [=====] - 5s 48ms/step - loss: 2.7448 - acc: 0.3108 - val_loss:

2.2657 - val_acc: 0.1933 Epoch 2/100

75/75 [=====] - 3s 42ms/step - loss: 1.0658 - acc: 0.6796 - val_loss:

2.6840 - val_acc: 0.2083 Epoch 3/100

75/75 [=====] - 3s 42ms/step - loss: 0.6163 - acc: 0.8308 - val_loss:

15.3506 - val_acc: 0.0933 Epoch 4/100

75/75 [=====] - 3s 43ms/step - loss: 0.4150 - acc: 0.9033 - val_loss:

2.1984 - val_acc: 0.4783 Epoch 5/100

75/75 [=====] - 3s 43ms/step - loss: 0.1654 - acc: 0.9563 - val_loss:

1.6799 - val_acc: 0.6233 Epoch 6/100
75/75 [=====] - 3s 42ms/step - loss: 0.1309 - acc:
0.9675 - val_loss:
1.9683 - val_acc: 0.6033 Epoch 7/100
75/75 [=====] - 3s 42ms/step - loss: 0.0753 - acc:
0.9796 - val_loss:
1.8538 - val_acc: 0.6950 Epoch 8/100
75/75 [=====] - 3s 44ms/step - loss: 0.0555 - acc:
0.9867 - val_loss:
0.3324 - val_acc: 0.9400 Epoch 9/100
75/75 [=====] - 3s 43ms/step - loss: 0.0451 - acc:
0.9912 - val_loss:
0.1921 - val_acc: 0.9817 Epoch 10/100
75/75 [=====] - 3s 43ms/step - loss: 0.0919 - acc:
0.9787 - val_loss:
2.4314 - val_acc: 0.6267 Epoch 11/100
75/75 [=====] - 3s 43ms/step - loss: 0.0571 - acc:
0.9850 - val_loss:
1.5475 - val_acc: 0.7600 Epoch 12/100
75/75 [=====] - 3s 43ms/step - loss: 0.0486 - acc:
0.9900 - val_loss:
0.3198 - val_acc: 0.9383 Epoch 13/100
75/75 [=====] - 3s 43ms/step - loss: 0.0305 - acc:
0.9900 - val_loss:
0.5827 - val_acc: 0.9267 Epoch 14/100
75/75 [=====] - 3s 42ms/step - loss: 0.0237 - acc:
0.9925 - val_loss:
0.5681 - val_acc: 0.9200 Epoch 15/100
75/75 [=====] - 3s 43ms/step - loss: 0.0478 - acc:
0.9908 - val_loss:
4.4333 - val_acc: 0.5933 Epoch 16/100
75/75 [=====] - 3s 42ms/step - loss: 0.0364 - acc:
0.9929 - val_loss:
0.7162 - val_acc: 0.8700 Epoch 17/100
75/75 [=====] - 3s 43ms/step - loss: 0.0192 - acc:
0.9950 - val_loss:
0.2874 - val_acc: 0.9767 Epoch 18/100
75/75 [=====] - 3s 43ms/step - loss: 0.0318 - acc:
0.9908 - val_loss:
0.4405 - val_acc: 0.9667 Epoch 19/100
75/75 [=====] - 3s 42ms/step - loss: 0.0071 - acc:
0.9987 - val_loss:

1.8619 - val_acc: 0.7683 Epoch 20/100
75/75 [=====] - 3s 42ms/step - loss: 0.0441 - acc:
0.9904 - val_loss:
1.1363 - val_acc: 0.7950 Epoch 21/100
75/75 [=====] - 3s 42ms/step - loss: 0.0399 - acc:
0.9933 - val_loss:
1.1070 - val_acc: 0.8167 Epoch 22/100
75/75 [=====] - 3s 42ms/step - loss: 0.0311 - acc:
0.9929 - val_loss:
0.3130 - val_acc: 0.9767 Epoch 23/100
75/75 [=====] - 3s 42ms/step - loss: 5.4799e-05 -
acc: 1.0000 - val_loss:
0.2564 - val_acc: 0.9750 Epoch 24/100
75/75 [=====] - 3s 42ms/step - loss: 0.0266 - acc:
0.9942 - val_loss:
0.2819 - val_acc: 0.9533
Epoch 25/100
75/75 [=====] - 3s 44ms/step - loss: 2.6892e-04 -
acc: 1.0000 - val_loss:
0.3835 - val_acc: 0.9533 Epoch 26/100
75/75 [=====] - 3s 42ms/step - loss: 0.0602 - acc:
0.9887 - val_loss:
2.1284 - val_acc: 0.7550 Epoch 27/100
75/75 [=====] - 3s 42ms/step - loss: 0.0195 - acc:
0.9950 - val_loss:
0.9641 - val_acc: 0.8600 Epoch 28/100
75/75 [=====] - 3s 42ms/step - loss: 0.0255 - acc:
0.9925 - val_loss:
0.6836 - val_acc: 0.9450 Epoch 29/100
75/75 [=====] - 3s 42ms/step - loss: 5.4592e-05 -
acc: 1.0000 - val_loss:
0.3799 - val_acc: 0.9800 Epoch 30/100
75/75 [=====] - 3s 42ms/step - loss: 0.0181 - acc:
0.9967 - val_loss:
0.3449 - val_acc: 0.9650 Epoch 31/100
75/75 [=====] - 3s 42ms/step - loss: 0.0447 - acc:
0.9900 - val_loss:
4.0432 - val_acc: 0.6600 Epoch 32/100
75/75 [=====] - 3s 45ms/step - loss: 0.0216 - acc:
0.9937 - val_loss:
0.7870 - val_acc: 0.9233 Epoch 33/100
75/75 [=====] - 3s 43ms/step - loss: 0.0099 - acc:

0.9992 - val_loss:
2.5992 - val_acc: 0.8217 Epoch 34/100
75/75 [=====] - 3s 42ms/step - loss: 0.0092 - acc:
0.9979 - val_loss:
2.4967 - val_acc: 0.8167 Epoch 35/100
75/75 [=====] - 3s 46ms/step - loss: 0.0293 - acc:
0.9942 - val_loss:
0.6163 - val_acc: 0.9633 Epoch 36/100
75/75 [=====] - 3s 42ms/step - loss: 3.2637e-05 -
acc: 1.0000 - val_loss:
0.4372 - val_acc: 0.9750 Epoch 37/100
75/75 [=====] - 3s 42ms/step - loss: 0.0380 - acc:
0.9929 - val_loss:
4.5018 - val_acc: 0.6567 Epoch 38/100
75/75 [=====] - 3s 42ms/step - loss: 2.4075e-05 -
acc: 1.0000 - val_loss:
13.0166 - val_acc: 0.4833 Epoch 39/100
75/75 [=====] - 3s 42ms/step - loss: 0.0458 - acc:
0.9925 - val_loss:
0.5825 - val_acc: 0.9333 Epoch 40/100
75/75 [=====] - 3s 42ms/step - loss: 0.0177 - acc:
0.9937 - val_loss:
1.3032 - val_acc: 0.8683 Epoch 41/100
75/75 [=====] - 3s 42ms/step - loss: 0.0378 - acc:
0.9933 - val_loss:
4.0145 - val_acc: 0.6700 Epoch 42/100
75/75 [=====] - 3s 42ms/step - loss: 0.0117 - acc:
0.9954 - val_loss:
1.2844 - val_acc: 0.8967 Epoch 43/100
75/75 [=====] - 3s 42ms/step - loss: 0.0227 - acc:
0.9962 - val_loss:
1.1683 - val_acc: 0.9000 Epoch 44/100
75/75 [=====] - 3s 42ms/step - loss: 0.0185 - acc:
0.9979 - val_loss:
0.5927 - val_acc: 0.9667 Epoch 45/100
75/75 [=====] - 3s 42ms/step - loss: 0.0493 - acc:
0.9946 - val_loss:
0.7778 - val_acc: 0.8933 Epoch 46/100
75/75 [=====] - 3s 42ms/step - loss: 0.0189 - acc:
0.9958 - val_loss:
0.3371 - val_acc: 0.9717 Epoch 47/100
75/75 [=====] - 3s 42ms/step - loss: 5.4999e-05 -

acc: 1.0000 - val_loss:
0.2090 - val_acc: 0.9883 Epoch 48/100
75/75 [=====] - 3s 42ms/step - loss: 0.0924 - acc:
0.9921 - val_loss:
0.3251 - val_acc: 0.9667 Epoch 49/100
75/75 [=====] - 3s 42ms/step - loss: 0.0223 - acc:
0.9958 - val_loss:
0.8978 - val_acc: 0.8883 Epoch 50/100
75/75 [=====] - 3s 42ms/step - loss: 0.0181 - acc:
0.9967 - val_loss:
5.0426 - val_acc: 0.6950 Epoch 51/100
75/75 [=====] - 3s 42ms/step - loss: 0.0059 - acc:
0.9987 - val_loss:
1.4630 - val_acc: 0.8750 Epoch 52/100
75/75 [=====] - 3s 42ms/step - loss: 0.0077 - acc:
0.9979 - val_loss:
8.0782 - val_acc: 0.5917 Epoch 53/100
75/75 [=====] - 3s 42ms/step - loss: 0.0118 - acc:
0.9975 - val_loss:
1.7128 - val_acc: 0.8500 Epoch 54/100
75/75 [=====] - 3s 43ms/step - loss: 0.0205 - acc:
0.9967 - val_loss:
11.3085 - val_acc: 0.5300 Epoch 55/100
75/75 [=====] - 3s 43ms/step - loss: 0.0186 - acc:
0.9950 - val_loss:
2.4782 - val_acc: 0.7350 Epoch 56/100
75/75 [=====] - 3s 42ms/step - loss: 0.0219 - acc:
0.9967 - val_loss:
0.5571 - val_acc: 0.9683 Epoch 57/100
75/75 [=====] - 3s 44ms/step - loss: 0.0125 - acc:
0.9979 - val_loss:
0.7435 - val_acc: 0.9633 Epoch 58/100
75/75 [=====] - 3s 42ms/step - loss: 0.0025 - acc:
0.9987 - val_loss:
0.8851 - val_acc: 0.9450 Epoch 59/100
75/75 [=====] - 3s 42ms/step - loss: 0.0182 - acc:
0.9967 - val_loss:
3.4652 - val_acc: 0.8117 Epoch 60/100
75/75 [=====] - 3s 42ms/step - loss: 0.0819 - acc:
0.9900 - val_loss:
1.2876 - val_acc: 0.9183 Epoch 61/100
75/75 [=====] - 3s 42ms/step - loss: 7.9829e-05 -

acc: 1.0000 - val_loss:
0.9401 - val_acc: 0.9400 Epoch 62/100
75/75 [=====] - 3s 42ms/step - loss: 0.0070 - acc:
0.9975 - val_loss:
1.9367 - val_acc: 0.8350 Epoch 63/100
75/75 [=====] - 3s 42ms/step - loss: 0.0022 - acc:
0.9992 - val_loss:
0.6993 - val_acc: 0.9650
Epoch 64/100
75/75 [=====] - 3s 42ms/step - loss: 5.0086e-06 -
acc: 1.0000 - val_loss:
0.5975 - val_acc: 0.9750 Epoch 65/100
75/75 [=====] - 3s 42ms/step - loss: 0.0153 - acc:
0.9971 - val_loss:
9.1949 - val_acc: 0.5867 Epoch 66/100
75/75 [=====] - 3s 42ms/step - loss: 0.0058 - acc:
0.9992 - val_loss:
0.4727 - val_acc: 0.9750 Epoch 67/100
75/75 [=====] - 3s 42ms/step - loss: 0.0203 - acc:
0.9971 - val_loss:
0.8094 - val_acc: 0.9650 Epoch 68/100
75/75 [=====] - 3s 42ms/step - loss: 2.5119e-06 -
acc: 1.0000 - val_loss:
0.5716 - val_acc: 0.9700 Epoch 69/100
75/75 [=====] - 3s 42ms/step - loss: 5.8429e-07 -
acc: 1.0000 - val_loss:
0.6544 - val_acc: 0.9683 Epoch 70/100
75/75 [=====] - 3s 42ms/step - loss: 0.0262 - acc:
0.9942 - val_loss:
0.8842 - val_acc: 0.9350 Epoch 71/100
75/75 [=====] - 3s 42ms/step - loss: 2.6946e-05 -
acc: 1.0000 - val_loss:
0.3366 - val_acc: 0.9750 Epoch 72/100
75/75 [=====] - 3s 42ms/step - loss: 0.0318 - acc:
0.9950 - val_loss:
0.9529 - val_acc: 0.9167 Epoch 73/100
75/75 [=====] - 3s 42ms/step - loss: 0.0010 - acc:
0.9996 - val_loss:
0.4976 - val_acc: 0.9750 Epoch 74/100
75/75 [=====] - 3s 42ms/step - loss: 2.2673e-06 -
acc: 1.0000 - val_loss:
0.7356 - val_acc: 0.9700 Epoch 75/100

75/75 [=====] - 3s 42ms/step - loss: 0.0122 - acc:
0.9962 - val_loss:
1.4073 - val_acc: 0.8817 Epoch 76/100
75/75 [=====] - 3s 45ms/step - loss: 0.2220 - acc:
0.9908 - val_loss:
7.6100 - val_acc: 0.7267 Epoch 77/100
75/75 [=====] - 3s 42ms/step - loss: 0.0249 - acc:
0.9958 - val_loss:
2.9434 - val_acc: 0.8083 Epoch 78/100
75/75 [=====] - 3s 42ms/step - loss: 2.3347e-05 -
acc: 1.0000 - val_loss:
0.4982 - val_acc: 0.9750 Epoch 79/100
75/75 [=====] - 3s 42ms/step - loss: 0.0038 - acc:
0.9996 - val_loss:
1.0316 - val_acc: 0.9317 Epoch 80/100
75/75 [=====] - 3s 42ms/step - loss: 3.0329e-06 -
acc: 1.0000 - val_loss:
0.4753 - val_acc: 0.9800 Epoch 81/100
75/75 [=====] - 3s 42ms/step - loss: 0.0528 - acc:
0.9942 - val_loss:
1.9978 - val_acc: 0.8333 Epoch 82/100
75/75 [=====] - 3s 42ms/step - loss: 0.0371 - acc:
0.9962 - val_loss:
8.6328 - val_acc: 0.6017 Epoch 83/100
75/75 [=====] - 3s 42ms/step - loss: 0.0011 - acc:
0.9996 - val_loss:
0.7949 - val_acc: 0.9667 Epoch 84/100
75/75 [=====] - 3s 42ms/step - loss: 0.0133 - acc:
0.9958 - val_loss:
4.4147 - val_acc: 0.8350 Epoch 85/100
75/75 [=====] - 3s 42ms/step - loss: 0.0065 - acc:
0.9992 - val_loss:
1.0086 - val_acc: 0.9700 Epoch 86/100
75/75 [=====] - 3s 42ms/step - loss: 1.6668e-06 -
acc: 1.0000 - val_loss:
0.6928 - val_acc: 0.9750 Epoch 87/100
75/75 [=====] - 3s 42ms/step - loss: 0.0150 - acc:
0.9987 - val_loss:
2.4998 - val_acc: 0.8667 Epoch 88/100
75/75 [=====] - 3s 42ms/step - loss: 0.0123 - acc:
0.9983 - val_loss:
2.6471 - val_acc: 0.8233 Epoch 89/100

```

75/75 [=====] - 3s 42ms/step - loss: 0.0033 - acc:
0.9987 -
val_loss:
3.1620 - val_acc: 0.8150 Epoch 90/100
75/75 [=====] - 3s 42ms/step - loss: 0.0031 - acc:
0.9996 - val_loss:
1.0187 - val_acc: 0.9517 Epoch 91/100
75/75 [=====] - 3s 42ms/step - loss: 0.0119 - acc:
0.9987 - val_loss:
0.7802 - val_acc: 0.9567 Epoch 92/100
75/75 [=====] - 3s 42ms/step - loss: 0.0025 - acc:
0.9987 - val_loss:
6.7926 - val_acc: 0.6817 Epoch 93/100
75/75 [=====] - 3s 42ms/step - loss: 0.0050 - acc:
0.9992 - val_loss:
0.6775 - val_acc: 0.9783 Epoch 94/100
75/75 [=====] - 3s 42ms/step - loss: 0.0180 - acc:
0.9975 - val_loss:
1.2644 - val_acc: 0.9533 Epoch 95/100
75/75 [=====] - 3s 42ms/step - loss: 0.0044 - acc:
0.9992 - val_loss:
3.5156 - val_acc: 0.7600 Epoch 96/100
75/75 [=====] - 3s 42ms/step - loss: 2.6273e-04 -
acc: 1.0000 - val_loss:
1.2655 - val_acc: 0.9533 Epoch 97/100
75/75 [=====] - 3s 42ms/step - loss: 0.0249 - acc:
0.9967 - val_loss:
4.4711 - val_acc: 0.8167 Epoch 98/100
75/75 [=====] - 3s 45ms/step - loss: 0.0302 - acc:
0.9950 - val_loss:
7.8894 - val_acc: 0.6800 Epoch 99/100
75/75 [=====] - 3s 45ms/step - loss: 0.0197 - acc:
0.9967 - val_loss:
1.6195 - val_acc: 0.8667 Epoch 100/100
75/75 [=====] - 3s 42ms/step - loss: 6.8973e-04 -
acc: 0.9996 - val_loss:
0.9372 - val_acc: 0.9700
<keras.callbacks.History at 0x7f4d5a680490>

```

Ek-2 için verilen görselleştirilmesi için ‘‘ matplotlib ‘‘ oluşturulan kod sistemi:

```

from matplotlib import pyplot
pyplot.figure(figsize=(4,4)) pyplot.plot(model.history.history['acc'])

```

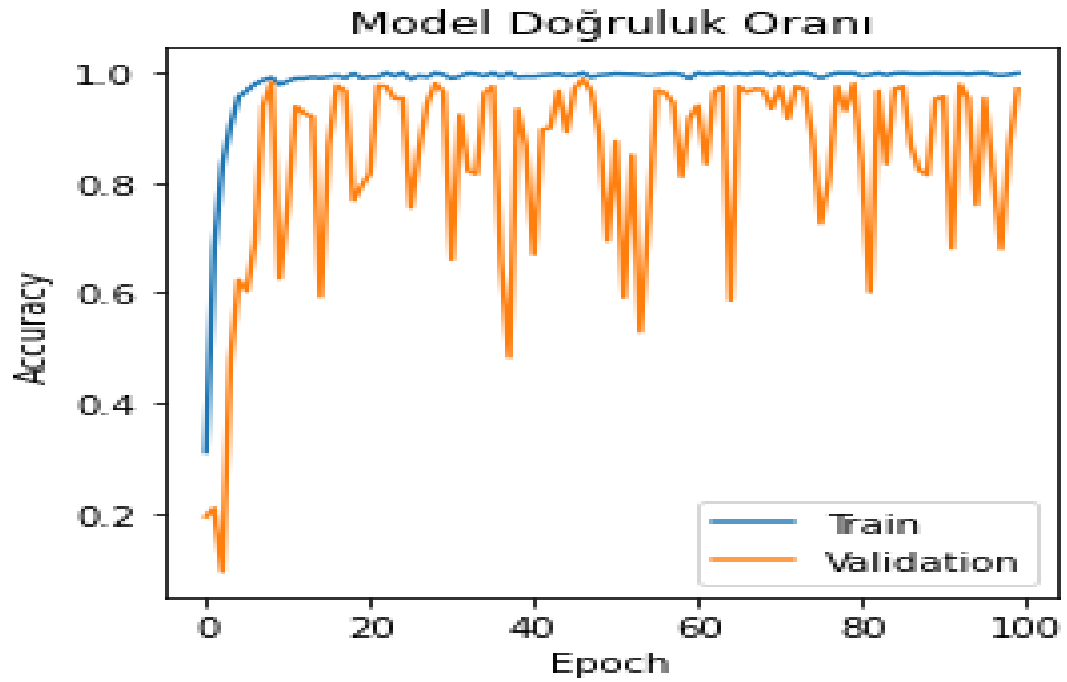
```

pyplot.plot(model.history.history['val_acc']) pyplot.title('Model Doğruluk Oranı')
pyplot.ylabel('Accuracy') pyplot.xlabel('Epoch')
pyplot.legend(['Train', 'Validation'], loc='lower right') pyplot.savefig('200.png',dpi=200)
print(max(model.history.history['val_acc']))

```

Grafik sonuç değeri ve görseli Ek.2:

0.9883333444595337



Şekil Ek-2.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi

```

model.save('model.h5')
from tensorflow.keras.models import load_model yuklenen_model =
load_model('model.h5')
from sklearn.metrics import confusion_matrix, classification_report
print(numpy.shape(Y))
tahminler = yuklenen_model.predict(X) tahminler = numpy.argmax(tahminler,axis=1)
print(numpy.shape(tahminler))
print(confusion_matrix(Y,tahminler)) print(classification_report(Y,tahminler))

```

Ek-2 için oluşan yukarıda ki kod betiğinin tablo çıktısı:

```

(1000, 1)
(1000,)
[[297  0  0  0  3  0  0  0  0  0]]

```

```

[ 0 300  0  0  0  0  0  0  0  0]
[ 0  0 297  0  0  0  0  0  3 0]
[ 0  0  0 300  0  0  0  0  0 0]
[ 0  0  0  0 300  0  0  0  0 0]
[ 0  3  0  0  0 297  0  0  0 0]
[ 0  3  0  3  0  0 294  0  0 0]
[ 0  0  0  0  0  0  0 300  0 0]
[ 0  0  0  3  0  0  0  0 297 0]
[ 0  0  0  0  0  0  0  0  0 300]]

```

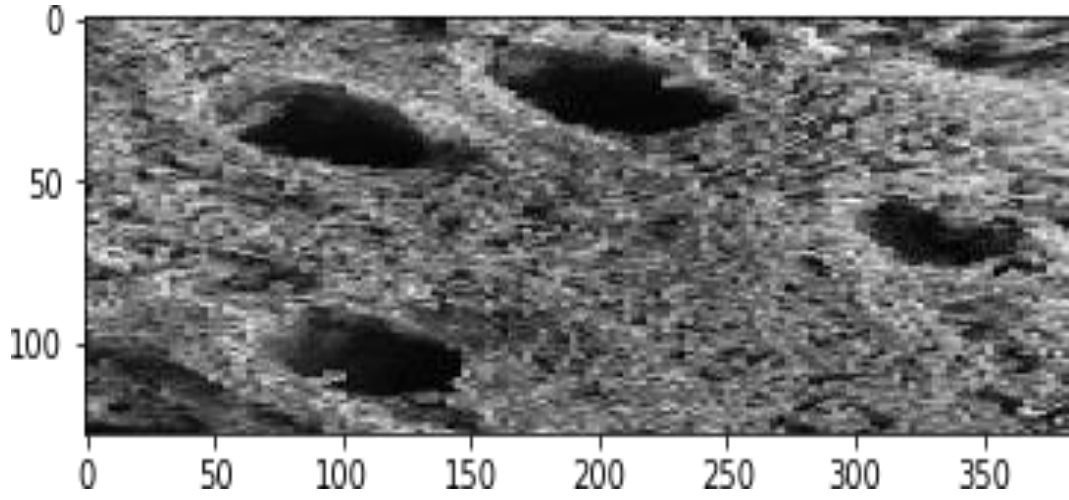
	precision	recall	f1-score	support
0	1.00	0.99	0.99	100
1	0.98	1.00	0.99	100
2	1.00	0.99	0.99	100
3	0.98	1.00	0.99	100
4	0.99	1.00	1.00	100
5	1.00	0.99	0.99	100
6	1.00	0.98	0.99	100
7	1.00	1.00	1.00	100
8	0.99	0.99	0.99	100
9	1.00	1.00	1.00	100
accuracy			0.99	1000
macro avg	0.99	0.99	0.99	1000
weighted avg	0.99	0.99	0.99	1000

Eđitim sonucunda oluřan řekil iin kod betiđi :
`print(tahminler[0]) pyplot.imshow(numpy.column_stack(X[100]),cmap='gray')`

Eđitim sonucunda, tahmini olarak oluřan řekil Ek.2:

3

<matplotlib.image.AxesImage at 0x7f4d4ba3dd90>



řekil Ek-2.2: AI tekniđi ile yapılan sınıflandırmada oluřan SEM(Lif) görüntü ıktısı

EK-3

RELU-softamax Adam = 0.000001 hızı ile ve Epoch = 25 değeri verilerek gerçekleştirilen kod dizini:

```
from tensorflow.keras import optimizers from tensorflow.keras import Sequential from
tensorflow import one_hot
from tensorflow.keras.callbacks import ModelCheckpoint from tensorflow.keras.layers
import Conv2D, Activation,
BatchNormalization, Flatten, Dense, MaxPool2D
model = Sequential()
model.add(Conv2D(filters=128,kernel_size=(3,3),input_shape=X[0].shape,ac
tivation='relu'))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(BatchNormalization()) model.add(Flatten())
model.add(Dense(64,activation='relu')) model.add(Dense(32,activation='relu'))
model.add(Dense(10,activation='softmax'))
print(numpy.shape(Y)) model.compile(loss='sparse_categorical_crossentropy',
optimizer=optimizers.Adam(learning_rate=0.000001), metrics=['acc'])
mc = ModelCheckpoint('en_ iyi.h5',save_best_only=True,monitor='val_loss',mode=
'min')
model.fit(X,Y,batch_size=32,epochs=25,shuffle=True,validation_split=0.2,
callbacks=[mc])
```

Ek-3 Sonuç:

```
(1000, 1)
Epoch 1/25
75/75 [=====] - 4s 47ms/step - loss: 2.2378 - acc:
0.1433 - val_loss:
2.2893 - val_acc: 0.1400 Epoch 2/25
75/75 [=====] - 3s 42ms/step - loss: 2.0726 - acc:
0.2146 - val_loss:
2.2641 - val_acc: 0.2033 Epoch 3/25
75/75 [=====] - 3s 43ms/step - loss: 1.9880 - acc:
0.2654 - val_loss:
2.2318 - val_acc: 0.2383 Epoch 4/25
75/75 [=====] - 3s 43ms/step - loss: 1.9167 - acc:
0.3008 - val_loss:
2.1841 - val_acc: 0.2800 Epoch 5/25
75/75 [=====] - 3s 43ms/step - loss: 1.8537 - acc:
0.3504 - val_loss:
```

2.1194 - val_acc: 0.3150 Epoch 6/25
75/75 [=====] - 3s 43ms/step - loss: 1.7899 - acc:
0.3950 - val_loss:
2.0353 - val_acc: 0.3550 Epoch 7/25
75/75 [=====] - 3s 43ms/step - loss: 1.7423 - acc:
0.4267 - val_loss:
1.9374 - val_acc: 0.3850 Epoch 8/25
75/75 [=====] - 3s 43ms/step - loss: 1.6785 - acc:
0.4596 - val_loss:
1.8385 - val_acc: 0.4133 Epoch 9/25
75/75 [=====] - 3s 43ms/step - loss: 1.6277 - acc:
0.4871 - val_loss:
1.7487 - val_acc: 0.4267 Epoch 10/25
75/75 [=====] - 3s 46ms/step - loss: 1.5876 - acc:
0.5133 - val_loss:
1.6749 - val_acc: 0.4450 Epoch 11/25
75/75 [=====] - 3s 44ms/step - loss: 1.5295 - acc:
0.5400 - val_loss:
1.6146 - val_acc: 0.4683 Epoch 12/25
75/75 [=====] - 3s 43ms/step - loss: 1.4847 - acc:
0.5579 - val_loss:
1.5649 - val_acc: 0.4867 Epoch 13/25
75/75 [=====] - 3s 43ms/step - loss: 1.4400 - acc:
0.5813 - val_loss:
1.5245 - val_acc: 0.4967 Epoch 14/25
75/75 [=====] - 3s 43ms/step - loss: 1.4021 - acc:
0.6033 - val_loss:
1.4823 - val_acc: 0.5267 Epoch 15/25
75/75 [=====] - 3s 43ms/step - loss: 1.3612 - acc:
0.6171 - val_loss:
1.4468 - val_acc: 0.5433 Epoch 16/25
75/75 [=====] - 3s 43ms/step - loss: 1.3192 - acc:
0.6333 - val_loss:
1.4102 - val_acc: 0.5733 Epoch 17/25
75/75 [=====] - 3s 43ms/step - loss: 1.2825 - acc:
0.6550 - val_loss:
1.3772 - val_acc: 0.5883 Epoch 18/25
75/75 [=====] - 3s 43ms/step - loss: 1.2490 - acc:
0.6717 - val_loss:
1.3457 - val_acc: 0.6050 Epoch 19/25
75/75 [=====] - 3s 43ms/step - loss: 1.2000 - acc:
0.6904 - val_loss:
1.3142 - val_acc: 0.6317 Epoch 20/25

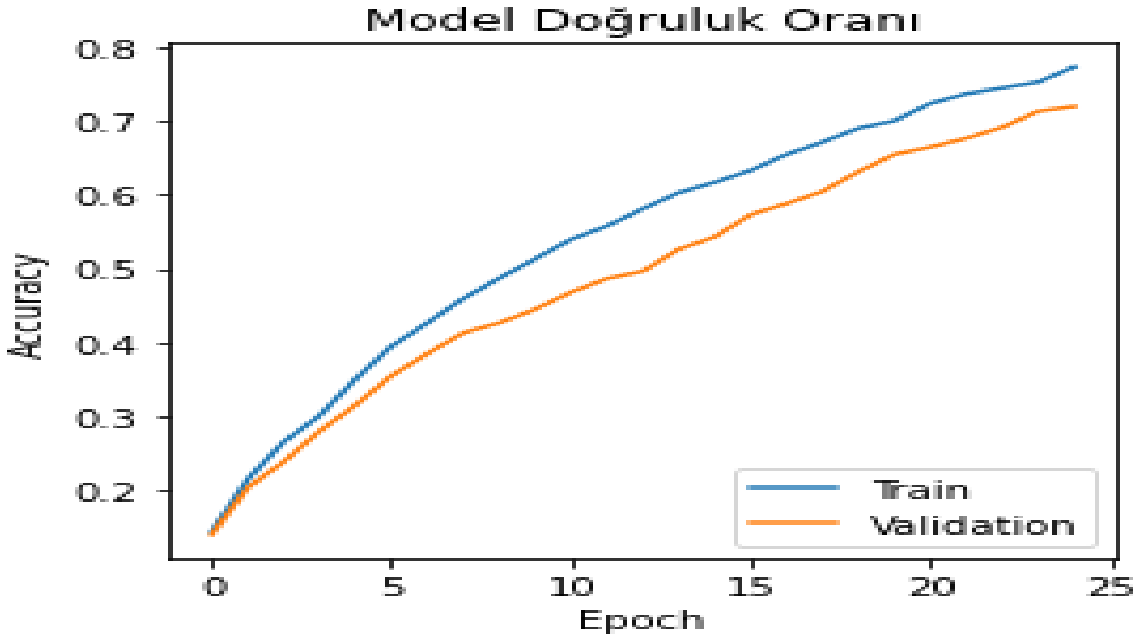
```
75/75 [=====] - 3s 43ms/step - loss: 1.1656 - acc:
0.7004 - val_loss:
1.2815 - val_acc: 0.6550 Epoch 21/25
75/75 [=====] - 3s 43ms/step - loss: 1.1326 - acc:
0.7242 - val_loss:
1.2531 - val_acc: 0.6650 Epoch 22/25
75/75 [=====] - 3s 43ms/step - loss: 1.1018 - acc:
0.7371 - val_loss:
1.2224 - val_acc: 0.6767 Epoch 23/25
75/75 [=====] - 3s 43ms/step - loss: 1.0709 - acc:
0.7450 - val_loss:
1.1960 - val_acc: 0.6917
Epoch 24/25
75/75 [=====] - 3s 45ms/step - loss: 1.0401 - acc:
0.7529 - val_loss:
1.1662 - val_acc: 0.7133 Epoch 25/25
75/75 [=====] - 3s 43ms/step - loss: 1.0120 - acc:
0.7738 - val_loss:
1.1411 - val_acc: 0.7200
<keras.callbacks.History at 0x7f4d4c7cebd0>
```

Ek-3 için verilen görselleştirilmesi için “ matplotlib ” oluşturulan kod sistemi:

```
from matplotlib import pyplot
pyplot.figure(figsize=(4,4)) pyplot.plot(model.history.history['acc'])
pyplot.plot(model.history.history['val_acc']) pyplot.title('Model Doğruluk Oranı')
pyplot.ylabel('Accuracy') pyplot.xlabel('Epoch')
pyplot.legend(['Train', 'Validation'], loc='lower right') pyplot.savefig('200.png',dpi=200)
print(max(model.history.history['val_acc']))
```

Grafik sonuç değeri ve görseli Ek.3:

0.7200000286102295



Şekil Ek-3.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan kod sistemi:

```
model.save('model.h5')
from tensorflow.keras.models import load_model
yuklenen_model = load_model('model.h5')
from sklearn.metrics import confusion_matrix, classification_report
print(numpy.shape(Y))
tahminler = yuklenen_model.predict(X)
tahminler = numpy.argmax(tahminler,axis=1)
print(numpy.shape(tahminler))
print(confusion_matrix(Y,tahminler))
print(classification_report(Y,tahminler))
```

Ek-3 için oluşan yukarıda ki kod betiğinin tablo çıktısı:

```
(1000, 1)
(1000,)
[[195  0  33  12  12  0  3  15  21 9]
 [ 12 249  6  9  0  0  3  0  18 3]
 [ 0  3 255  12  6  0  3  6  15 0]
 [ 15  3  15 198  12  3  6  12  36 0]
 [ 15  3  3  18 249  0  0  0  6 6]
```

```

[ 3  0 24 21  6 231  0  6  63]
[18  6 57  3  9  3 159  9 279]
[ 0  0  0  0  0  0  0 291  90]
[ 3 12  6  3  0  0  0 12 2640]
[ 6  6 12  0  0  0  0 15  0261]]

```

	precision	recall	f1-score	support
0	0.73	0.65	0.69	100
1	0.88	0.83	0.86	100
2	0.62	0.85	0.72	100
3	0.72	0.66	0.69	100
4	0.85	0.83	0.84	100
5	0.97	0.77	0.86	100
6	0.91	0.53	0.67	100
7	0.80	0.97	0.87	100
8	0.66	0.88	0.75	100
9	0.90	0.87	0.88	100
accuracy			0.78	1000
macro avg	0.80	0.78	0.78	1000
weighted avg	0.80	0.78	0.78	1000

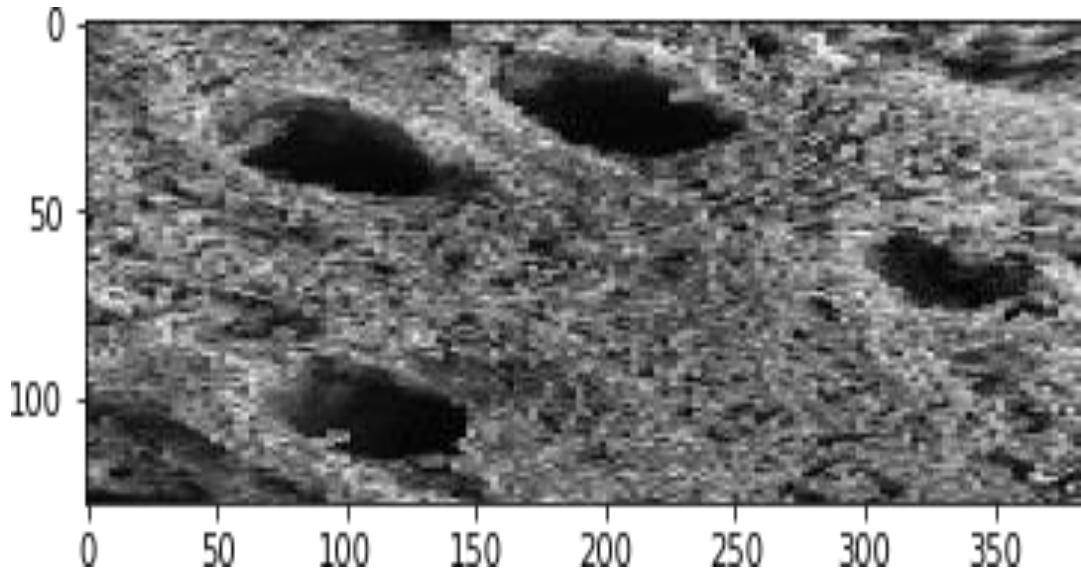
Eğitim sonucunda oluşan şekil için kod betiği :

```
print(tahminler[0]) pyplot.imshow(numpy.column_stack(X[100]),cmap='gray')
```

Eğitim sonucunda, tahmini olarak oluşan Şekil Ek.3:

3

<matplotlib.image.AxesImage at 0x7f4d4ba3dd90>



Şekil Ek-3.2: AI tekniği ile yapılan sınıflandırmada oluşan SEM(Lif) görüntü çıktısı

EK-4

RELU-softamax Adam = 0.000001 hızı ile ve Epoch = 100 değeri verilerek gerçekleştirilen kod dizini:

```
from tensorflow.keras import optimizers from tensorflow.keras import Sequential from
tensorflow import one_hot
from tensorflow.keras.callbacks import ModelCheckpoint from tensorflow.keras.layers
import Conv2D, Activation,
BatchNormalization, Flatten, Dense, MaxPool2D
model = Sequential()
model.add(Conv2D(filters=128,kernel_size=(3,3),input_shape=X[0].shape,activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(BatchNormalization()) model.add(Flatten())
model.add(Dense(64,activation='relu')) model.add(Dense(32,activation='relu'))
model.add(Dense(10,activation='softmax'))
print(numpy.shape(Y)) model.compile(loss='sparse_categorical_crossentropy',
optimizer=optimizers.Adam(learning_rate=0.000001), metrics=['acc'])
mc = ModelCheckpoint('en_yyi.h5',save_best_only=True,monitor='val_loss',mode='min')
model.fit(X,Y,batch_size=32,epochs=25,shuffle=True,validation_split=0.2,
callbacks=[mc])
```

Ek-4 Sonuç:

```
(1000, 1)
Epoch 1/25
75/75 [=====] - 4s 47ms/step - loss: 2.2378 - acc:
0.1433 - val_loss:
2.2893 - val_acc: 0.1400 Epoch 2/25
75/75 [=====] - 3s 42ms/step - loss: 2.0726 - acc:
0.2146 - val_loss:
2.2641 - val_acc: 0.2033 Epoch 3/25
75/75 [=====] - 3s 43ms/step - loss: 1.9880 - acc:
0.2654 - val_loss:
2.2318 - val_acc: 0.2383 Epoch 4/25
75/75 [=====] - 3s 43ms/step - loss: 1.9167 - acc:
0.3008 - val_loss:
2.1841 - val_acc: 0.2800 Epoch 5/25
75/75 [=====] - 3s 43ms/step - loss: 1.8537 - acc:
```

0.3504 - val_loss:
2.1194 - val_acc: 0.3150 Epoch 6/25
75/75 [=====] - 3s 43ms/step - loss: 1.7899 - acc:
0.3950 - val_loss:
2.0353 - val_acc: 0.3550 Epoch 7/25
75/75 [=====] - 3s 43ms/step - loss: 1.7423 - acc:
0.4267 - val_loss:
1.9374 - val_acc: 0.3850 Epoch 8/25
75/75 [=====] - 3s 43ms/step - loss: 1.6785 - acc:
0.4596 - val_loss:
1.8385 - val_acc: 0.4133 Epoch 9/25
75/75 [=====] - 3s 43ms/step - loss: 1.6277 - acc:
0.4871 - val_loss:
1.7487 - val_acc: 0.4267 Epoch 10/25
75/75 [=====] - 3s 46ms/step - loss: 1.5876 - acc:
0.5133 - val_loss:
1.6749 - val_acc: 0.4450 Epoch 11/25
75/75 [=====] - 3s 44ms/step - loss: 1.5295 - acc:
0.5400 - val_loss:
1.6146 - val_acc: 0.4683 Epoch 12/25
75/75 [=====] - 3s 43ms/step - loss: 1.4847 - acc:
0.5579 - val_loss:
1.5649 - val_acc: 0.4867 Epoch 13/25
75/75 [=====] - 3s 43ms/step - loss: 1.4400 - acc:
0.5813 - val_loss:
1.5245 - val_acc: 0.4967 Epoch 14/25
75/75 [=====] - 3s 43ms/step - loss: 1.4021 - acc:
0.6033 - val_loss:
1.4823 - val_acc: 0.5267 Epoch 15/25
75/75 [=====] - 3s 43ms/step - loss: 1.3612 - acc:
0.6171 - val_loss:
1.4468 - val_acc: 0.5433 Epoch 16/25
75/75 [=====] - 3s 43ms/step - loss: 1.3192 - acc:
0.6333 - val_loss:
1.4102 - val_acc: 0.5733 Epoch 17/25
75/75 [=====] - 3s 43ms/step - loss: 1.2825 - acc:
0.6550 - val_loss:
1.3772 - val_acc: 0.5883 Epoch 18/25
75/75 [=====] - 3s 43ms/step - loss: 1.2490 - acc:
0.6717 - val_loss:
1.3457 - val_acc: 0.6050 Epoch 19/25
75/75 [=====] - 3s 43ms/step - loss: 1.2000 - acc:
0.6904 - val_loss:

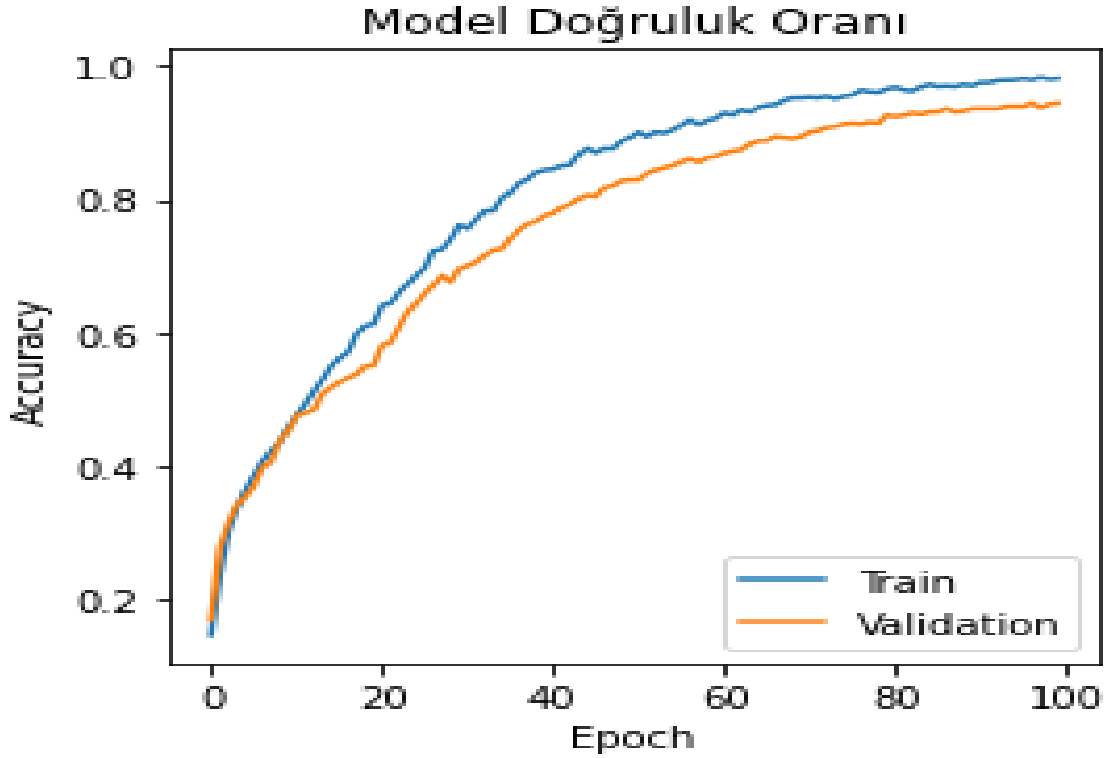
```
1.3142 - val_acc: 0.6317 Epoch 20/25
75/75 [=====] - 3s 43ms/step - loss: 1.1656 - acc:
0.7004 - val_loss:
1.2815 - val_acc: 0.6550 Epoch 21/25
75/75 [=====] - 3s 43ms/step - loss: 1.1326 - acc:
0.7242 - val_loss:
1.2531 - val_acc: 0.6650 Epoch 22/25
75/75 [=====] - 3s 43ms/step - loss: 1.1018 - acc:
0.7371 - val_loss:
1.2224 - val_acc: 0.6767 Epoch 23/25
75/75 [=====] - 3s 43ms/step - loss: 1.0709 - acc:
0.7450 - val_loss:
1.1960 - val_acc: 0.6917
Epoch 24/25
75/75 [=====] - 3s 45ms/step - loss: 1.0401 - acc:
0.7529 - val_loss:
1.1662 - val_acc: 0.7133 Epoch 25/25
75/75 [=====] - 3s 43ms/step - loss: 1.0120 - acc:
0.7738 - val_loss:
1.1411 - val_acc: 0.7200
<keras.callbacks.History at 0x7f4d4c7cebd0>
```

Ek-4 için verilen görselleştirilmesi için ‘‘ matplotlib ‘‘ oluşturulan kod sistemi:

```
from matplotlib import pyplot
pyplot.figure(figsize=(4,4)) pyplot.plot(model.history.history['acc'])
pyplot.plot(model.history.history['val_acc']) pyplot.title('Model Doğruluk Oranı')
pyplot.ylabel('Accuracy') pyplot.xlabel('Epoch')
pyplot.legend(['Train', 'Validation'], loc='lower right') pyplot.savefig('200.png',dpi=200)
print(max(model.history.history['val_acc']))
```

Grafik sonuç değeri ve görseli Ek.4:

0.9449999928474426



Şekil Ek-4.1: Değerlerin tensorflow.keras ile modellenerek; rapor şekline getirmek için oluşturulan

kod sistemi:

```
model.save('model.h5')
from tensorflow.keras.models import load_model yuklenen_model =
load_model('model.h5')
from sklearn.metrics import confusion_matrix, classification_report
print(numpy.shape(Y))
tahminler = yuklenen_model.predict(X) tahminler = numpy.argmax(tahminler,axis=1)
print(numpy.shape(tahminler))
print(confusion_matrix(Y,tahminler)) print(classification_report(Y,tahminler))
```

Ek-4 için oluşan yukarıda ki kod betiğinin tablo çıktısı:

(1000, 1)

(1000,)

```
[[291  0  0  0  0  0  3  0  3 3]
 [ 3 297  0  0  0  0  0  0  0 0]
 [ 0  0 279  9  0  0  0  0  9 3]
 [ 0  0  0 300  0  0  0  0  0 0]
 [ 0  0  0  3 297  0  0  0  0 0]
 [ 0  0  0  6  0 294  0  0  0 0]
 [ 0  3  0  3  3  0 285  0  3 3]
 [ 0  0  0  0  0  0  0  0 300  0]
 [ 0  0  0  3  0  0  0  0 297 0]
 [ 0  0  0  0  0  0  0  3  0 297]]
```

	precision	recall	f1-score	support
0	0.99	0.97	0.98	100
1	0.99	0.99	0.99	100
2	1.00	0.93	0.96	100
3	0.93	1.00	0.96	100
4	0.99	0.99	0.99	100
5	1.00	0.98	0.99	100
6	0.99	0.95	0.97	100
7	0.99	1.00	1.00	100
8	0.95	0.99	0.97	100
9	0.97	0.99	0.98	100
accuracy			0.98	1000
macro avg	0.98	0.98	0.98	1000
weighted avg	0.98	0.98	0.98	1000

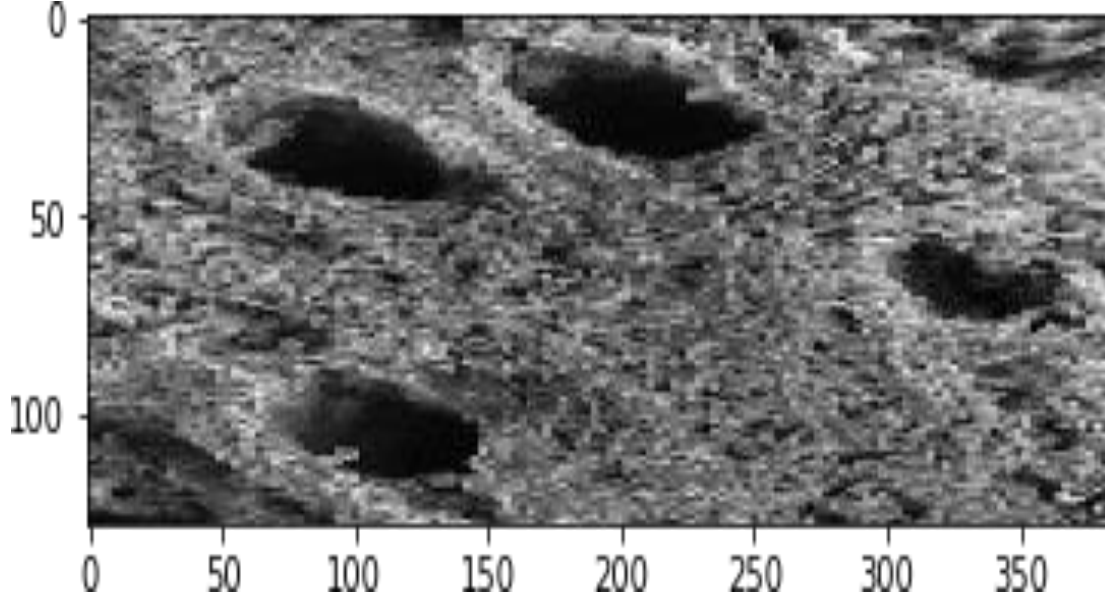
Eğitim sonucunda oluşan şekil için kod betiği :

```
print(tahminler[0]) pyplot.imshow(numpy.column_stack(X[100]),cmap='gray')
```

Eđitim sonucunda, tahmini olarak oluřan Őekil Ek.4:

3

<matplotlib.image.AxesImage at 0x7f4d48c36c10>



Őekil Ek-4.2: AI tekniđi ile yapılan sınıflandırmada oluřan SEM(Lif) görüntü çıktıısı

ÖZGEÇMİŞ

Adı-Soyadı : Ayşe DEMİRKAN

Eğitim ve Mesleki Geçmişi:

- 2021-2023 ALKÜ, Mühendislik Fakültesi, Metalürji ve Malzeme Mühendisliği, Nanobilim ve Nanomühendislik Tezli yüksek lisans eğitimi.
- 2017-2020, NKU, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- 2013-2015, DEU, Atatürk İlke ve İnkılap Tarihi, (Tezli YLS)
- 2011-2015, AÜ, Sosyoloji (YLS)
- 2010- Halen MEB Matematik Öğretmenliği
- 2002-2006, Eğitim Fakültesi, İlköğretim Matematik Öğretmenliği/Fen Bilgisi Öğretmenliği (Çift Anadal)

Meslek yaşamına MEB’de matematik öğretmeni olarak devam etmektedir. 2017’de yılında başladığı Bilgisayar Mühendisliğini 2020’de NKU bitirip mezun oldu. Bölümü okurken yapay zekâ ile ilgili çeşitli kurslara ve eğitimlere ve konferanslara katılıp R,WEKA,Docker,MATLAB ve Python’da uzmanlaşmış;’’Thresholding’’,’’Iris Recognition’’ ile ilgili kod çalışmaların gerçekleştirdiği sunumlarının yanı sıra; bitirme tezini ‘‘Yapay zekâ ve tıp alanında gelişmeler’’ üzerine yazmıştır.

Erasmus projeleri kapsamında BAU-GOETHE university ortaklığı ile 2016 yılında gerçekleşen ‘‘BOOM!’’ projesinde aktif rol almıştır.

2013-2015 yıllarında DEU ‘nde ‘‘Matematik Eğitimi’’ üzerine yazdığı tez ile tezli yüksek lisansını tamamlamıştır.

2011-2015 yılları arasında İlber ORTAYLI’nın ‘‘Her eğitimcinin sosyoloji bitirmesi gereklidir’’ sözüne istinaden Anadolu Üniversitesi’nde Sosyoloji eğitimini tamamlamıştır.

2002-2009 çeşitli kurumlarda çalışmış ve ardından 2010’da kadrolu olarak göreve başlamıştır. 2002-2006 OMÜ ilköğretim matematik/Fen bilimleri alanında çift diploma alarak mezun olmuştur.

2022 yılında Tübitak kapsamında Deneyap Nanoteknoloji eğitimliği yapmıştır. 2022 yılında tezin oluşum ve yazım aşamasında, altı konferansta ‘‘SEM görüntülerinin Yapay Zekâ eğitimi’’ sunumunu gerçekleştirmiş ve başarı sertifikası ile katılım belgelerini alarak; sunumları altı ulusal ve uluslararası kitapta özet ve tam metin olarak

yayımlanmıştır. EJOSAT dergisinde bir makalesi yayımlanmıştır.

Ayrıca çeşitli Avrupa/ABD yayımlarının önde gelen dergileri başta olmak üzere; uluslararası bilimsel dergilere makale yazma aşamasındadır.

Bilgisayar mühendisi olarak Colab ve Github'da yazılım geliştirip; proje oluşumları için freelance ve online çalışma fırsatlarını değerlendirmektedir.

Doktora hazırlık aşamasında olan Ayşe DEMİRKAN Nanobilim ve Nanomühendislik yüksek lisansını 2021'de ALKÜ'de başlayıp; ders aşamasını tamamlayıp; yaklaşık bir buçuk yıldır ‘SEM görüntülerinin yapay zekâ metotları ile incelenmesi’ üzerine deneysel çalışmalarını sürdürdüğü tezini oluşturmaktadır.