

**T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ**

**DUYGU TANIMA İÇİN DERİN ÖĞRENME
MODELLERİNİN KARŞILAŞTIRILMASI VE
UYGULANABİLİRLİĞİ**

Mesut UYSAL

Danışman: Doç. Dr. Mehmet Fatih DEMİRAL

BURDUR, 2024

ETİK KURALLARA UYGUNLUK BEYANI

Burdur Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili hükümleri uyarınca **Yüksek Lisans Tezi** olarak sunduğum “**Duygu Tanıma İçin Derin Öğrenme Modellerinin Karşılaştırılması ve Uygulanabilirliği**” başlıklı bu tezin;

- Kendi çalışmam olduğunu,
- Sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi,
- Bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi,
- Kullandığım verilerde değişiklik yapmadığımı,
- Tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını,
- Bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı,
- Bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı,

bildirir, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

03/07/2024

(İmza)

Mesut UYSAL

ÖNSÖZ

Bugünlere gelmemde maddi ve manevi her türlü konuda destek olan bana güvenlerini hiç eksik etmeyen tüm aileme sonsuz saygı ve sevgilerimi sunarım.

Bu tez çalışmasında yüksek lisans öğrenimim boyunca yanımda olan, deneyim ve bilgisiyle karşılaştığım her problemde destek olan ve yol gösteren değerli danışman hocam **Doç. Dr. Mehmet Fatih DEMİRAL’ a** sonsuz teşekkür ederim.

Lisans eğitimim boyunca bu alanda gelişmemde rol oynayan, desteklerini hiçbir zaman esirgemeyen kıymetli hocalarım **Prof. Dr. Ali Hakan IŞIK** ve **Prof. Dr. Emre ÇOMAK’ a** teşekkürlerimi sunarım.

Çalışma boyunca veri seti temininde ve tez sürecinde takıldığım noktalarda yardımcı olan değerli abim **Mesut BİLGİN’ e** şükranlarımı sunarım.

Bugünlere gelmemde maddi ve manevi her türlü konuda destek olan, bana güvenlerini hiç eksik etmeyen tüm aileme sonsuz saygı ve sevgilerimi sunarım.

Temmuz, 2024

Mesut UYSAL

İÇİNDEKİLER

	Sayfa
ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
ÇİZELGELER DİZİNİ	v
SİMGELER VE KISALTMALAR DİZİNİ	vi
ÖZET	vii
ABSTRACT.....	viii
1. GİRİŞ	1
2. GENEL BİLGİLER	3
2.1. Literatür Taraması	3
2.2. Yapay Zekâ	14
2.3. Makina Öğrenmesi	15
2.3.1. Denetimli Öğrenme	16
2.3.2. Denetimsiz Öğrenme	16
2.3.3. Pekiştirmeli Öğrenme	17
2.4. Yapay Sinir Ağları (YSA).....	17
2.4.1. Yapay Sinir Ağı Katmanları	19
2.4.2. Yapay Sinir Ağlarının Temel Yapısı	20
2.4.3. Aktivasyon Fonksiyonları.....	21
2.5. Derin Öğrenme	24
2.5.1. Derin Öğrenmede Kullanılan YSA Modelleri.....	25
2.5.2. Derin Öğrenmede Transfer Öğrenme (Pre-Trained Modeller).....	31
3. MATERYALLER	33
3.1. Python.....	33
3.2. VS Stüdyo	33
3.3. Google Colab.....	34
3.4. Kütüphaneler	35
3.4.1. Pandas	35
3.4.2. NumPy	35
3.4.3. Matplotlib	36
3.4.4. Seaborn	36
3.4.5. Scikit-learn.....	36
3.4.6. Keras:	36
3.4.7. TensorFlow:.....	37
3.4.8. Kivy	37
3.5. Veri.....	37
3.6. Yöntemler.....	40
3.6.1. DenseNet121.....	40
3.6.2. EfficientNet.....	46
3.6.3. MobileNetV2	53
3.6.4. ResNet50.....	59
3.6.5. VGG16.....	65
3.6.6. YOLOv8m-cls	69
4. BULGULAR	72
5. SONUÇ VE ÖNERİLER	77
KAYNAKLAR	79

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Yapay zekâ ve alt dalları.....	14
Şekil 2.2. Makine öğrenmesi çeşitleri.....	16
Şekil 2.3. Yapay sinir ağı katmanları.....	19
Şekil 2.4. Sinir hücresinin bileşenleri	20
Şekil 2.5. Doğrusal fonksiyon grafiği.....	21
Şekil 2.6. Sigmoid fonksiyon gösterimi.....	22
Şekil 2.7. Softmax Fonksiyon grafiği	23
Şekil 2.8. ReLU fonksiyon grafiği.....	23
Şekil 2.9. Tanh fonksiyon grafiği.	24
Şekil 2.10. RNN katman yapısı	26
Şekil 2.11. CNN yapısı	27
Şekil 2.12. Evrişim katmanında matris işlemi	28
Şekil 2.13. Havuzlama katmanının gösterimi.....	29
Şekil 2.14. <i>Flatten</i> katmanda düzleştirme işlemi.....	29
Şekil 2.15. Dropout katmanı gösterimi.....	30
Şekil 2.16. Toplu normalleştirme örneği.	30
Şekil 3.1. Veri seti örnek görseller.....	38
Şekil 3.2. Veri seti modele hazırlama aşaması	40
Şekil 3.3. DenseNet-121'in şeması	41
Şekil 3.4. DenseNet-121'e eklenen katman şeması.....	42
Şekil 3.5. DenseNet-121 accuracy değerleri.....	44
Şekil 3.6. DenseNet-121 kayıp değerleri	44
Şekil 3.7. DenseNet-121 karmaşıklık matrisi	45
Şekil 3.8. DenseNet-121 ROC eğrisi	46
Şekil 3.9. EfficientNet-B5 şeması.....	48
Şekil 3.10. EfficientNet-B5 eklenen katman şeması	49
Şekil 3.11. EfficientNet-B5 accuracy değerleri	51
Şekil 3.12. EfficientNet-B5 kayıp değerleri	51
Şekil 3.13. EfficientNet-B5 karmaşıklık matrisi	52
Şekil 3.14. EfficientNet-B5 ROC eğrisi	53
Şekil 3.15. MobilenetV2 model şeması	54

Şekil 3.16. MobilenetV2 eklenen katman şeması.....	55
Şekil 3.17. MobilenetV2 accuracy değerleri	56
Şekil 3.18. MobilenetV2 loss değerleri	56
Şekil 3.19. MobilenetV2 ROC eğrisi.....	58
Şekil 3.20. MobilenetV2 karmaşıklık matrisi	58
Şekil 3.21. ResNet50 model şeması.....	60
Şekil 3.22. ResNet50 eklenen model şeması	62
Şekil 3.23. ResNet50 accuracy grafiği.....	63
Şekil 3.24. ResNet50 Kayıp Grafiği	63
Şekil 3.25. ResNet50 karmaşıklık matrisi.....	64
Şekil 3.26. ResNet50 ROC grafiği.....	65
Şekil 3.27. VGG16 model şeması.....	66
Şekil 3.28. VGG16 accuracy grafiği.....	67
Şekil 3.29. VGG16 kayıp grafiği	67
Şekil 3.30. VGG16 karmaşıklık matrisi.....	68
Şekil 3.31. VGG16 ROC eğrisi	68
Şekil 3.32. YOLOv8m-cls karmaşıklık matrisi	70
Şekil 4.1. Uygulama çalışma şeması.....	72
Şekil 4.2. EfficientNet GPU ve sistem RAM kullanımı	74
Şekil 4.3. MobilNetV2 GPU ve sistem RAM kullanımı	75
Şekil 4.4. VGG16 GPU ve sistem RAM kullanımı	75
Şekil 4.5. DenseNet GPU ve sistem RAM kullanımı	75
Şekil 4.6. ResNet GPU ve Sistem RAM kullanımı	76
Şekil 4.7. Yolov8m-cls GPU ve sistem RAM kullanımı	76

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1. Literatür özeti.....	13
Çizelge 3.1. Veri seti sınıf ve miktar tablosu.....	39
Çizelge 3.2. DenseNet-121 derleme aşamasındaki kullanılan parametreler.....	43
Çizelge 3.3. DenseNet-121 eğitim parametreleri ve callbackler	44
Çizelge 3.4. DenseNet-121 metrik tablosu	45
Çizelge 3.5. EfficientNet-B5 derleme aşamasındaki kullanılan parametreler	50
Çizelge 3.6. EfficientNet-B5 eğitim aşamasında parametreler.....	50
Çizelge 3.7. EfficientNet-B5 metrik tablosu.....	51
Çizelge 3.8. MobilenetV2 derleme aşamasında kullanılan parametreler	55
Çizelge 3.9. MobilenetV2 eğitim aşamasında kullanılan parametreler	56
Çizelge 3.10. MobilenetV2 metrik tablosu	59
Çizelge 3.11. ResNet50 derleme aşamasında kullanılan parametreler	62
Çizelge 3.12. ResNet50 eğitim aşamasında kullanılan parametreler.....	63
Çizelge 3.13. ResNet50 metrik tablosu.....	64
Çizelge 3.14. VGG16 derleme aşamasında kullanılan parametreler	66
Çizelge 3.15. VGG16 Eğitim aşamasında kullanılan parametreler	66
Çizelge 3.16. VGG16 metrik tablosu.....	68
Çizelge 3.17. YOLOv8m-cls değer tablosu.....	71
Çizelge 4.1. Karşılaştırmalı sonuç değer tablosu.....	73

SİMGELER VE KISALTMALAR DİZİNİ

API	: Uygulama geliştirme arabirimi
BiLSTM	: İki yönlü LSTM (Bidirectional LSTM)
CNN	: Evrişimli Sinir ağı
CPU	: Merkezi İşlem birimi
DBOW	: Dağıtık Kelime Torbaları (Distributed Bag of Words)
DPM	: Deformable Part Model
DSÖ	: Dünya Sağlık Örgütü
DT	: Decision Tree (Karar Ağaçları)
DVM	: Destek Vektör Makineleri
FPR	: Yanlış Pozitif Oranı
GPU	: Grafiksel İşleme Birimi (Graphics Processing Unit)
IDE	: Entegre Geliştirme Ortamı
KNN	: En yakın komşu algoritması
LR	: Lineer Regresyon
LR	: Logistic Regression
LSTM	: Uzun Süreli Hafıza Yöntemi
ML	: Machine Learning
MLP	: Çok katmanlı algılayıcı
NB	: Naive Bayes
ReLU	: Doğrultulmuş Lineer Birim (Rectified Linear Unit)
RF	: Random Forest
RNN	: Yinelemeli Sinir Ağı
RO	: Rassal Orman
ROC	: Receiver Operating Characteristic eğrisi
SGD	: Stokastik Gradyan İniş
SVM	: Destek Vektör Makineleri
TF	: TensorFlow
TF-IDF	: Term Frequency-Inverse Document Frequency
TPR	: Doğru Pozitif Oranı
YSA	: Yapay Sinir Ağı
YZ	: Yapay Zeka

ÖZET

Yüksek Lisans Tezi

Duygu Tanıma İçin Derin Öğrenme Modellerinin Karşılaştırılması ve Uygulanabilirliği

Mesut Uysal

Burdur Mehmet Akif Ersoy Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mehmet Fatih DEMİRAL

Temmuz, 2024

Bu tezde, çeşitli yüz ifadelerini tanımlamak ve sınıflandırmak için derin öğrenme mimarileri kullanılarak duygu tespiti alanında bir çalışma yapılmıştır. Bu çalışmanın amacı, öfke, küçümseme, iğrenme, korku, mutluluk, tarafsızlık, üzüntü ve şaşkınlık gibi farklı yüz ifadelerini ayırt etmektir. Çalışma, DenseNet121, EfficientNetB5, MobileNetV2, ResNet50, VGG16 ve YOLOv8m-cls gibi bilgisayarla görme ve makine öğrenimi alanında farklı özellik ve yeteneklere sahip iyi bilinen modellerin kullanım nedenlerini ve bu modellerin karşılaştırmalı sonuçlarını içermektedir. Araştırma süresince, çeşitli duygusal ifadeler içeren veri setleri titizlikle hazırlanmış ve kapsamlı model eğitimi ve değerlendirmesi sağlanmıştır. Veri setleri, özellik çıkarma sürecini iyileştirmek ve potansiyel önyargıları azaltmak için ön işleme tabi tutulmuştur. Daha sonra, her model yüz ifadelerini doğru bir şekilde tespit etmek ve sınıflandırmak için performanslarını optimize edecek gerekli yöntemlerle eğitilmiştir. Sonuçlar, her mimarinin doğruluğunu ve zayıf yönlerini açığa çıkararak, gerçek dünya senaryolarındaki uygulanabilirliklerini aydınlatmıştır. Bulgular, belirli kullanım durumları ve hesaplama kısıtlamaları için uygun modellerin seçilmesinin önemini vurgulamaktadır. Gelecekte, bu araştırma, duygu tespiti tekniklerinin daha fazla araştırılmasını ve geliştirilmesini destekleyerek sağlık hizmetleri, insan-bilgisayar etkileşimi ve duygusal bilişim gibi çeşitli alanlarda faydalar sağlayabilir.

Verilen modeller arasında en yüksek doğruluk değerine %89 ile YoloV8m-cls modeli erişirken, en kötü performans ve en büyük kayıplar ise %60 doğruluk ile VGG16 modeli gösterdiği için uygun bir yöntem değildir.

Anahtar Kelimeler: derin öğrenme, duygu tespiti, yüz ifadeleri, model karşılaştırması, transfer öğrenme.

ABSTRACT

M.Sc. Thesis

Comparative Analysis and Applicability of Deep Learning Models for Emotion Recognition

Mesut Uysal

**Burdur Mehmet Akif Ersoy University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering**

Supervisor: Assoc. Prof. Dr. Mehmet Fatih Demiral

July, 2024

In this thesis, a study was conducted in the field of emotion detection by utilizing deep learning architectures to identify and classify various facial expressions. The aim of this study is to distinguish between different facial expressions such as anger, contempt, disgust, fear, happiness, neutrality, sadness, and surprise. The study includes the reasons for using well-known models with different features and capabilities in the field of computer vision and machine learning, such as DenseNet121, EfficientNetB5, MobileNetV2, ResNet50, VGG16, and YOLOv8m-cls, along with their comparative results. Throughout the research, datasets containing various emotional expressions were meticulously prepared to provide comprehensive model training and evaluation. The datasets were pre-processed to enhance the feature extraction process and reduce potential biases. Subsequently, each model was trained using necessary methods to optimize their performance in accurately detecting and classifying facial expressions. The results exposed the accuracy and weaknesses of each architecture, shedding light on their applicability in real-world scenarios. The findings highlight the importance of selecting models that are suitable for specific use cases and computational constraints. In the future, this research has the potential to support further investigation and development of emotion detection techniques, providing benefits in various fields such as healthcare, human-computer interaction, and affective computing.

Among the given models, the YoloV8m-cls model achieved the highest accuracy with 89%, while the VGG16 model showed the worst performance and the largest losses with an accuracy of 60%, making it an unsuitable method.

Keywords: deep learning, emotion detection, facial expressions, model comparison.

1. GİRİŞ

Duygu analizi, insan iletişiminin derinliklerine inen ve iletişimin karmaşıklığının anlaşılmasına yardımcı olan bir araçtır. Her gün milyonlarca insanın farklı duygular içeren mesajlar gönderdiği bir dünyada, bu mesajların arkasındaki gerçek anlamı ve niyeti anlamak önemlidir. Duygu analizi; metin, ses veya görüntü gibi iletişim biçimlerini inceleyerek, bunların içerdiği duyguları tanımlamak, anlamak ve yorumlamak için kullanılan bir yöntemdir. İletişimdeki bu duygusal derinliği anlamak, birçok alanda son derece önemlidir. Söz gelimi pazarlama kampanyaları duygusal tepkileri hedef alarak daha etkili hâle getirebilir, müşteri memnuniyeti üzerinde olumlu etkiler yaratabilir ve hatta toplumsal olayların ve trendlerin öngörülmesine yardımcı olabilir. Dolayısıyla duygu analizi, iletişimin altında yatan duygusal katmanları anlamak için kritik bir araçtır. Bu da insan etkileşimlerinin daha derinlemesine anlaşılmasına ve insanların etkilenmesine yardımcı olur.

Duygu analizi, günümüzde derin öğrenme modellerinin sağladığı olanaklarla önemli bir gelişme göstermiştir. Bu gelişme, insan-bilgisayar etkileşimi, duygusal hesaplama ve sağlık sektörü gibi geniş bir uygulama yelpazesine sahiptir. Derin öğrenme ve yapay zekâ teknolojileri; metin, ses ve görüntü verilerini analiz ederek bunların içerdiği duyguları belirlemek ve yorumlamak için giderek daha karmaşık ve hassas yöntemler geliştirmiştir. Özellikle sosyal medya platformları gibi büyük veri kaynakları, duygusal içeriklerin analizi için zengin bir veri havuzu sağlamak ve bu durum duygu analizindeki ilerlemeyi hızlandırmaktadır.

Bireylerin karşılaştığı çeşitli zorluk ve baskılar, özellikle aile baskısı, depresyon kaygısı, ekonomik sıkıntılar gibi faktörler, bireylerin duygularını ifade etmekte güçlük çekmelerine neden olabilir. Toplumun bu tür bireylere yardımcı olabilmesi ve destek sağlayabilmesi için onların duygusal durumlarını doğru bir şekilde anlamak kritik öneme sahiptir. İşte bu noktada duygu analizi çalışmaları büyük bir önem kazanmaktadır. Örneğin, bir sosyal medya platformunda kullanıcıların paylaştığı içeriklerin duygu analizi yapılarak toplumun genel duygusal durumu hakkında önemli bilgiler elde edilebilir ve bu bilgiler, duygusal destek hizmetleri gibi önemli kaynakların daha etkin bir şekilde yönlendirilmesine yardımcı olabilir. Ayrıca duygu analizi, psikolojik yardım ve danışmanlık hizmetlerinin kişiselleştirilmesinde de önemli bir rol oynayabilir, çünkü bu veriler bireylerin duygusal durumlarını daha iyi anlamak ve yanıt vermek için kullanılabilir. Bu nedenle duygu analizi, modern toplumun karşılaştığı duygusal zorluklarla başa çıkma ve duygusal refahı artırma konusunda kritik bir araç hâline gelmiştir.

Duygu analizi, insanların duygusal durumlarını deęerlendirmek, tepkilerini anlamak ve duygusal zekâ arařtırmalarına katkı saęlamak gibi amalar tařımaktadır. Özellikle duygusal zorluklarla mcadele eden bireyler iin duygusal analizlerin yapılması ve duygusal saęlıklarının anlařılması byk nem tařımaktadır. Bu tr analizler, bireylerin duygu dnyalarının daha iyi anlařılmasına yardımcı olabilir ve gerektięinde destek saęlamak iin uygun stratejiler geliřtirilmesine olanak tanır.

Bu baęlamda, bu alıřma eřitli derin ęrenme modellerinin kapsamlı bir řekilde incelenmesi ve karřılařtırılmasıyla duygu tanıma alanına katkı sunmayı amalamaktadır. Temel ama; fke, kmseme, ięrenme, korku, mutluluk, tarafsızlık, znt ve řařkınlık gibi insan duygularını belirleyen yz ifadelerini doęru bir řekilde tanımlamak ve sınıflandırmaktır. alıřma bu ynyle duygusal ifadelerin doęru bir řekilde tanınması, duygusal durumları deęerlendirmek, insanların tepkilerini anlamak ve duygusal zekâ arařtırmaları gibi birok alana katkı saęlayabilir.

Bu tez alıřması farklı derin ęrenme mimarilerinin duygu tanıma performansını kapsamlı bir řekilde deęerlendirmeyi hedeflemektedir. Bu amala eřitli metrikler kullanılarak her modelin performansı llmektedir. Bu metrikler arasında doęruluk, hassasiyet, F1 puanı gibi standart performans ltleri bulunmaktadır. Ancak bunlarla sınırlı kalmayarak ayrıca duygu tanıma modellerinin eęitim sreleri, hiper parametre optimizasyonları ve veri n iřleme adımları gibi faktrler de detaylı bir řekilde incelenmektedir. Her bir modelin eęitim sreci titizlikle analiz edilerek hangi mimari ve yaklařımın duygu tanıma grevinde en etkili olduęunu belirlemek iin kapsamlı bir karřılařtırma yapılacaktır.

Derin ęrenme modelleriyle yapılan duygu analizi alıřmaları, bu alandaki teknolojik ilerlemeleri ve analitik yetenekleri kullanarak insan duygularını doęru bir řekilde tanımlama ve sınıflandırma konusunda byk bir potansiyel sunmaktadır. Bu analizler, duygusal ifadelerin yanı sıra duygusal durumların objektif bir řekilde deęerlendirilmesine yardımcı olabilir. Özellikle belirli duygusal zorluklarla bařa ıkmak iin geliřtirilen modeller, bireylerin duygusal ihtiyalarını anlamak ve uygun destekleri saęlamak adına nemli bir ara olabilir.

Sonuç olarak duygu analizi alıřmaları, derin ęrenme modellerinin sunduęu olanakları kullanarak insanların duygusal saęlıęını anlamaya ve iyileřtirmeye ynelik nemli bir adımdır. Bu alıřmaların geniřletilmesi ve daha kapsamlı bir řekilde deęerlendirilmesi, duygusal zorluklarla bařa ıkmak ve daha saęlıklı bir duygusal yařam srdrmek isteyen bireyler iin deęerli bir kaynak olabilir.

2. GENEL BİLGİLER

2.1. Literatür Taraması

Yılmaz ve Orman (2021) yaptıkları araştırmada Twitter’da yazılan metinleri kullanarak duygu analizi üzerine çalışmışlardır. Duygu analizi, metinleri çeşitli bilgisayar algoritmalarının sınıflandırmasıyla gerçekleştirilen bir işlemdir. Çalışmada Türkçe ifadeler toplanarak olumlu ve olumsuz olmak üzere iki gruba ayrılmıştır. Uzun Süreli Hafıza (LSTM) yönteminin kullanıldığı bu çalışmada veri kümelerinde %97 doğruluk elde edilmiştir.

Şahiner Yılmaz vd. (2021) tarafından yapılan çalışmada metin verilerinden derin öğrenme teknikleri kullanılarak duyguların ortaya çıkarılması amaçlanmıştır. Bu çalışmada 2 Türkçe veri seti kullanılmıştır. İlki 1 milyon üzerinde ve yaklaşık 5 GB boyutu ile yer kaplayan Twitter verisi, ikincisi ise bir yarışmadan alınan 31 binin üzerindeki veridir. Eldeki veriler saldırgan ve saldırgan değil şeklinde iki gruba ayrılmıştır. Çalışmada Uzun Kısa Dönemli Bellek (LSTM) ve Çift Yönlü Uzun Kısa Dönemli Bellek (BiLSTM) ağları kullanarak sınıflandırma performansları ölçülmüştür. LSTM modelinde doğruluk değeri %86, duyarlılık değeri %55, kesinlik %68, F- skor %61 bulunurken BiLSTM modelinde doğruluk %86, duyarlılık %55, kesinlik %66 ve F-skor %60 sonuçları elde edilmiştir. Bunların yanında derin sinir ağları da mukayese edilmiş ve en başarılı modelin LSTM olduğu sonucuna varılmıştır.

Kilimeci (2020) tarafından gerçekleştirilen çalışmanın konusu Borsa İstanbul 100 endeksinin yönünün tahmin edilmesidir. Haber kaynağı olarak Twitter platformunu kullanan yazar sosyal ağlarda fikirlerin ifade edilmesinde boyut kısıtlaması olduğundan sınıflandırma performansının bundan önemli ölçüde etkilendiğini ifade etmektedir. Bu yüzden araştırma, veri kümelerinin anlamsal açıdan çeşitli yöntemlerle zenginleştirilip topluluk öğrenmesi yaklaşımını derin öğrenme algoritmalarıyla harmanlayarak sınıflandırma performansını iyileştirmeyi hedeflemektedir. Bu çalışmanın çeşitli özellikleri sebebiyle alanın ilk ve tek çalışması olduğu iddia edilmiştir. Örneğin bu araştırmada performans artırmak için herkese açık Türkçe ve İngilizce Twitter veri kümeleri kullanılmıştır. Karar verme yöntemlerinde nihai kararı vermek için çoğunluk oylaması ve model birleştirme teknikleri kullanılmıştır. Yine tek bir öğrenme algoritması yerine CNN, RNN ve LSTM gibi derin öğrenme algoritmalarının harmanlanmasıyla derin topluluk modeli kullanılmıştır. Çalışma sonucunda Türkçe veri kümesi için %78,07 İngilizce veri kümesi için ise %78,92 sınıflandırma sonucu elde edilmiştir.

Kaynar vd. (2017) tarafından yapılan çalışmada duygu analizinde boyut düşürme tekniğinin etkisi ölçülmüştür. IMDB film yorumlarından oluşan veri seti YSA ve DVM sınıflandırma algoritmaları kullanılarak ilk boyutları ile eğitilmiş ve ikinci adımda boyutları düşürülmüştür. Ön işlem işlemi gerçekleştirilmiştir. Son adımda eldeki düşük boyuttaki veriler sisteme verilerek eğitime aşaması sağlanmıştır. Sonuçlar değerlendirildiğinde boyutlar düşürüldükten sonra daha çok başarı sağlandığı görülmüştür. Fakat çTBA tekniği kullanılarak veri seti üzerinde bir başarı sağlanmadığı ve bu tekniğin bu veri seti için uygun bir yöntem olmadığı belirtilmiştir.

Bilgin ve Şentürk (2019) tarafından yapılan çalışma, eldeki veri setinin yarı danışmanlı ve danışmanlı eğitimin veri seti üzerindeki etkisini araştırmaya yöneliktir. Çalışmada doküman vektörünün 2 farklı yöntemi olan DBoW ve DM yöntemlerinin performansları değerlendirilmiştir. Çalışmada hem Türkçe hem de İngilizce veriler üzerinde duygusal sınıflandırma yapılmıştır. Veri seti ön işlemeden sonra eğitim için veriler, test aşamasına hazırlanmış ve dört farklı makine öğrenmesi tekniği (doğruluk, kesinlik, duyarlılık, F-ölçütü) için listelenmiştir. Araştırma sonuçlarına bakıldığında yarı danışmanlı eğitimin performans ölçütlerinin daha başarılı olduğu görülmektedir. Ayrıca DBoW yöntemi DM yöntemine göre daha iyi sonuç vermektedir. DM yönteminin geri kalmasının sebeplerinden biri olarak daha fazla veriye ihtiyaç duyulduğundan eldeki verinin yetersiz kalması olduğu düşünülmektedir. Çalışma sonunda gelecek çalışmalarda daha fazla Türkçe veri seti ile çalışılması önerilmekte olup etiketli veri sayısının yükseltilmesi hedeflenmiştir.

Çalışkan ve Demir (2022) tarafından gerçekleştirilen bu çalışmada derin öğrenme yöntemleriyle şüpheli davranış tespiti üzerine bir araştırma yapılmıştır. Marmara Üniversitesi Kriminal Davranış / Nesne veri setinden alınan 1116 etiketli görselden alınan veriler, YOLOv4 gerçek zamanlı veri algılama algoritması ile çalışılmıştır. Derin öğrenmede eğitim verisinin çok olması model başarımına etki ettiğinden veriler %70 eğitim verisi, %30 test verisi olarak ayrılmıştır. YOLOv4 modelinin tasarımı ve yapısı belirlenmiştir, sonra veriler bu algorithmada eğitilmektedir. Eğitim sonunda %86,6'lık bir başarıya ulaşılmıştır. Bu çalışmada sadece izni olmayan veriler üzerinden şüphe tespiti yapılırken hırsızlık, sahtecilik gibi durumlar ile ilgili nesne tespiti yapıldığı takdirde yüksek başarıya ulaşılacağı tahmin edilmektedir.

Açıkalın vd., (2020) tarafından yapılan çalışmada film ve otel yorumlarından oluşan veri seti iki ayrı sınıfa (pozitif ve negatif) ayrılarak iyi bir başarı elde edilmiştir. Çalışma boyunca BERT modeli kullanılmıştır. Beyazperde.com'dan alınan film yorumları 5 ayrı gruba ayrılarak değerlendirilmiştir. 1-2 puan alan yorumlar negatif, 4-5 puan alan yorumlar ise pozitif olarak değerlendirilmiştir. Bu şekilde 26.700 pozitif ve 26.700 negatif puan mevcuttur. Yine otel için

otelpuan.com'dan alınan veriler 0-100 arası değerlendirilmiş, 0-40 arası bir puana sahip model negatif, 80-100 arası puan alan bir yorum ise pozitif olarak ölçülmüştür. Toplam 11.600 yorumdan 5.800 yorumun negatif sınıfa, 5.800 yorumun ise pozitif sınıfa ait olduğu bulgusuna varılmıştır.

Şahinaslan vd. (2022), Naive Bayes sınıflandırma algoritması kullanarak sosyal medya verileri üzerinden dünya dillerinde yapılan yorum ve paylaşımları incelemeyi hedeflemişlerdir. Youtube yorumları kullanılan bu çalışmada Türkçe, İngilizce, Almanca, Arapça, Rusça ve Korece dillerine ait 15082 veri kullanılmıştır. Yorumlar Naive Bayes algoritması kullanılarak pozitif, negatif ve nötr olarak sınıflandırılmış ve WEKA ile yapılan analizde doğru sınıflandırma %65,56 olarak bulunmuştur. Çalışma sırasında veriler PHP ile temizlenip Google çeviri aracı API'si ile İngilizceye çevrilmiştir. Google çeviri aracı 108 farklı dilde destek sağlamıştır.

Akdeniz (2022) tarafından yapılan makine öğrenmesi alanındaki çalışma, sosyal medya verilerinin duygu analizi alanında nasıl kullanılacağı ve makine öğrenmesi yöntemlerinin bu verileri nasıl işleyeceğine yöneliktir. Mart 2022 tarihinde tüm dünyayı ve ülkemizi etkileyen koronavirüsten dolayı uzaktan eğitime geçildiği dönemde Twitter'da paylaşılan Türkçe karakterli yazılar ve tweetler bu çalışmanın veri setini oluşturmaktadır. Öncelikle eldeki veri seti bir ön işleme tabi tutulmuş, bu sayede imla ve yazı hataları giderilerek veri daha anlaşılır hâle getirilmiştir. Bu işlem zemberek kütüphanesinden faydalanılarak yapılmıştır. Bundan sonra ise veri seti için manuel etiketleme işlemi yapılmış, dil çevirisi yöntemiyle İngilizceden doğrudan duygu çıktıları üreten modeller oluşturulmuştur. Bu çalışma içerisinde farklı sayısallaştırma yöntemleri (BoW, TF-IDF, Word2Vec) ve makine öğrenme modelleri (LR, SGD, SVM, RF, NB) kullanılmıştır. Performans sonuçlarına bakıldığında Türkçe metinlerin manuel etikete sahip olduğu en iyi TF-IDF ve LR ikilisi 0,79'luk bir sınıflandırma başarısı gösterirken manuel yöntemle etiketlenenmiş metinler veri setinden çıkarıldığında başarı oranının yükseldiği ve BoW-LR ikilisinin 0,84'lük sonuç verdiği görülmektedir.

Köksal vd., (2021) tarafından gerçekleştirilen çalışmada Twitter'da duygu analizi yöntemiyle Bitcoin'de değer tahminleme işlemi ele alınmıştır. Araştırmada, Twitter kullanıcılarının Bitcoin ile ilgili yorumları derlenerek bir duygu analizi çalışması gerçekleştirilmiştir. Kullanıcı yorumları, Twitter'ın API hizmeti aracılığıyla Python programlama dili kullanılarak olumlu, nötr ve olumsuz sınıflara ayrılmıştır. Naive Bayes ve Lojistik Regresyon algoritmaları kullanılarak oluşturulan modellerin başarı oranları karşılaştırılmıştır. Naive Bayes uygulamasının duygulardaki doğru tahmin etme oranı %72,19, lojistik regresyon uygulamasında ise bu oran %75,53 olarak bulunmuştur. 2. bölümde ise

“Bitcoin” anahtar kelimesini içeren günlük pozitif tweet oranı ile Bitcoin günlük açılış değeri beraber kullanılarak Bitcoin kapanış değeri tahmin edilmiştir. Finans değerleri Yahoo Finance web sitesinden alınmış ve Doğrusal Regresyon ile Rastgele Orman Regresyon yöntemleriyle değerlendirilmiştir. Doğrusal Regresyon r^2 değeri %88,97 Rastgele Orman Regresyonu için ise %94,16 olarak bulunmuştur.

Göçgün ve Onan (2021) tarafından Amazon ürünlerinin değerlendirilmesi üzerinde derin öğrenme ve makine öğrenmesi teknikleri kullanılarak duygu analizi ele alınmıştır. Amazon ürünlerine yapılan yorumların veri setini oluşturduğu bu çalışmada makine öğrenmesiyle yorumların duygu içerip içermediği, içeriyorsa da üç farklı sınıf oluşturularak (olumlu, olumsuz, tarafsız) hangi duyguyu içerdiğinin tespit edilmesi amaçlanmıştır. Kaggle’den alınan Amazon yorumları veri seti, Python dilinde Jupyter Lab ortamında 4 farklı makine öğrenmesi yoluyla işlenmiştir. Sırasıyla bu yöntemler karar ağacı algoritması, Naive Bayes, Destek Vektör Makineleri(SVM) ve Lojistik Regresyondur. Çalışma sonunda en başarılı yöntem, %94 doğruluk ve %75 F-skor puanıyla Karar Ağacı yöntemi çıkarken başarı olarak en düşük makine öğrenmesi modeli Naives Bayes olarak saptanmıştır.

Öztürk ve Pashaei (2021) tarafından konuşma ve ses telleri ile duyguları analiz etme işlemi üzerine çalışılmıştır. Bu çalışmanın temeli insanların ses ve konuşmaların şeklinden duyguları hissediyor olmasına karşın makinelerin bu konuda pek başarılı olamadığından hareketle duygu analizi makineler üzerinden de denenmiştir. Bu sebeple çalışmada CNN (Evrışım Sinir Ağları) mimarisi ve Uzun-Kısa Süreli Bellek (LSTM) yöntemleri kullanılmıştır. CNN model için 3 çekirdek boyutunda dört farklı nöronlu katman (128, 256, 256, 512) kullanılmıştır. CNN model, seslerin öznitelik çıkarımını yaparken LSTM ile de bu öznitelikler birbirleriyle ilişkilendirilmiştir. Ayrıca çalışmada MFCC (Mel Frequency Cepstral Coefficient) kullanılarak konuşma sinyallerinden görüntü öznitelik matrisi çıkarılmış ve boyut indirgeme işlemi yapılmıştır. Sınıflandırma yöntemi olarak Co-LSTM kullanılmış olup performans sonuçlarına göre %86,7’lik bir doğruluk sonucuna ulaşılmıştır.

Öge ve Kayaalp (2021) tarafından yapılan çalışmada teknolojik imkânların ve sosyal medya mecrasının kullanımının artmasıyla beraber ortaya çıkan işlenmemiş veriler kullanılarak 6 farklı sınıflandırma algoritmasıyla duygu analizi işlemi gerçekleştirilmiştir. Python programlama dili kütüphanelerinden faydalanılarak Yapay Sinir Ağları (ANN), Rastgele Orman (RF), Naive Bayes (NB), Lojistik Regresyon (LR), Destek Karar Makinası (SVM) ve Karar Ağacı (DT) algoritmalarıyla işlenen verilerin performansları birbirleriyle kıyaslanmıştır. Ayrıca veri setinin sayısallaştırma işlemi için Bag of Word (BoW), TF-IDF, FastText ve Word2Vec gibi dört farklı temsil yöntemi kullanılmıştır. Bu çalışma için IMDB

sitesinden alınan 50.000 film yorumu kullanılmıştır. Tüm yorumlar tek bir dosyada toplanarak Jupyter Notebook ortamında Pandas kütüphanesi yardımıyla verilerin okunması sağlanmıştır. Kullanılan bu 50.000 verinin 25 .000'i pozitif, 25.000'i ise negatif yorumdan oluştuğundan veri seti gayet düzgün bir dağılıma sahip olmaktadır. Eğitim öncesi rastgele karıştırılan veri seti beş ayrı gruba ayrılarak %80 eğitim, %20 test verisi olacak şekilde eğitilmiştir. Eğitim sonunda her bir yöntem doğruluk, kesinlik, duyarlılık ve f1-skor metrikleri ile ölçülmüş ve en başarılı sonuç Lojistik Regresyon ile Destek Vektör Makinası olarak bulunmuştur. Çalışma hızı açısından ise en hızlı word2vec, en yavaş ise Destek Vektör Makinası sonucu elde edilmiştir.

Turan vd. (2022) tarafından son zamanlarda ortaya çıkan sanal evren kavramı ile ilgili kullanıcıların yaptıkları yorumlar kaynak alınarak duygu tespiti çalışması yapılmıştır. Veri seti olarak Youtube ve Twitter'da yapılan yorumların ele alındığı bu çalışmada veriler Naive Bayes, Rastgele Orman, Lojistik Regresyon ve Destek Vektör Makine algoritmalarında (SVM) işlenmiş ve sonuçlar birbirleriyle mukayese edilmiştir. Eğitim öncesi bozuk ve hatalı veriler ayrıştırılmış, Youtube'dan 1350 veri, Twitter'dan 1350 veri eğitime tabi tutulmuştur. Performans değerleri, doğruluk ve F-skor metrikleri ile değerlendirilmiştir. Her iki metrik dört farklı sınıflandırma algoritması ile ölçüldüğünde tüm değerlerde %80 üzerinde başarımlar sağlanırken en başarılı model %88 ile Naive Bayes algoritması olarak bulunmuştur. Ayrıca çalışmada sanal evren hakkında Twitter'da yapılan yorumların %46'sı olumlu bulunurken Youtube'da yapılan yorumların %55'i olumlu olarak öne çıkmaktadır. Genel ortalamaya bakıldığında sanal evren hakkında iki kaynaktaki yorumların %50,4'ü olumlu, %49,6'sının olumsuz olduğu sonucuna varılmıştır.

Safalı (2021) tarafından gerçekleştirilen çalışmada dört farklı veri tabanından alınan 360.000 görüntü üzerinde derin öğrenme tekniklerinden VGG-16, AlexNet, ZFNet mimarileri kullanılarak yüz görüntülerinden duygu analizi işlemi yapılmıştır. Bu çalışma için sırasıyla celebA 180.000, FFHQ 50.000, LFW 10.000 ve Youtube'dan 120.000 veri alınarak veri seti oluşturulmuştur. Eldeki verilerin 300.000'i eğitim için, 60.000'i test için ayrılmıştır. Görüntü verileri yedi farklı grupta (iğrenme, üzgün, korkak, mutlu, sinirli, nötr, şaşkın) sınıflandırılmıştır. Eldeki veriler 244x244x1 boyutunda olup eğitim öncesi kırılarak ve yeniden boyutlandırılarak eğitilmiştir. Öncelikle her model onar kez eğitilmiş, ilk tur sonunda VGG-16 %55,77, AlexNet %47,35, ZF Net %41,11 başarımlar sağlanmıştır. VGG-16 mimarisi 43. turdan sonra sabit kalmaya devam ederken AlexNet mimarisinin 38. turdan sonra ve ZF Net mimarisinin de 46. turdan sonra başarı oranı değişmemiştir. Eğitim sonrası en başarılı model %92,03 ile VGG-16 olmuştur.

Zilyas ve Yılmaz (2023) tarafından yapılan çalışmada ortaokul öğrencilerinin başarımının ölçülmesi ve öğrenmelerini olumsuz etkileyen faktörlerin makine öğrenme algoritmaları aracılığıyla belirlenmesi amaçlanmıştır. 246 erkek, 273 kız öğrenciden oluşan toplam 519 öğrencinin on üçer sorudan oluşan bir ankete verdikleri cevaplar doğrultusunda oluşturulan veri seti 13 sütun, 520 satırdan oluşmaktadır. Ankette cinsiyet, kardeş sayısı, özel eğitim alıp almama durumu, anne ve babanın ayrı/birlikte olduğunu tespitiye yönelik 13 soru yöneltilmiştir. Veri seti; %80 eğitim, %20 test şeklinde ayrıldıktan sonra eğitim öncesi yüksek başarım elde edilmesi için veri temizleme ve ön işleme adımlarından geçirildikten sonra Rastgele Orman, Çoklu Lineer Regresyon, Gradyan Arttırıcı Regresyon, KNN, Torbalanmış Ağaçlar, Karar Ağaçları algoritmalarında eğitim işlemi gerçekleştirilmiştir. Çıktı sonuçlarına göre en başarılı makine öğrenme algoritması %98 ile Rastgele Orman algoritması ve %95 ile R-kare skor bulunmuştur.

Ayan vd., (2019) tarafından yapılan çalışmada Twitter yorumlarından hareketle tweetlerin İslamafobi içerip içermediği tespit edilmiştir. Makine öğrenme algoritması olarak Lineer Regresyon ve Naive Bayes sınıflandırıcıları ile eğitilen modellerin F1-skor, duyarlılık, regresyon metrikleri ile performansları ölçülmüştür. Çalışmada verilerin bir arada bulundurulması için MYSQL veri tabanından faydalanılmıştır. İngilizce anahtar kelimelerle İslam ile ilgili 162.000 tweet çekilmiş ve bu tweetler pozitif ve negatif olarak sınıflandırılmıştır. Çekilen verilerdeki Twitter harici linkler silinmiş, benzer olan ifadelerden birisi elenerek veri seti eğitim öncesi temizleme ve ön işlemeye alınmıştır. Veri setindeki yorumların %51'i olumlu, %49'u ise olumsuz ifadelerden oluşmaktadır. Eğitim için %80 eğitim, %20 test şeklinde ayrılan veriler sonucunda F1 ölçütüne göre Ridge regresyonunda %96,9 Naive Bayes algoritmasında ise %95,4'lük başarım elde edilmiştir. Eğitim süresi açısından ise Ridge regresyonunda daha kısa eğitim işlemi gerçekleştirilmiştir.

Uyaroğlu Akdeniz ve Cebeci (2021) çalışmalarında Sakarya ili merkezli olmak üzere Adapazarı, Serdivan, Erenler ve Hendek belediyeleri hakkında olumlu ve olumsuz yorumlardan belediye hizmetinin memnuniyetini ölçmüşlerdir. Bu çalışmada derin öğrenme tekniklerinden Turkish Bert yöntemi kullanılmıştır. Bu çalışmadaki temel amaç halkın belediye hakkındaki olumlu veya olumsuz düşüncesini saptayarak istek ve şikayetlere daha hızlı dönüş yapabilmektir. Sakarya Büyükşehir ve adı geçen ilçelerden alınan 10.000 Türkçe tweet, bu çalışmanın veri setini oluşturmaktadır. Veri seti içindeki # gibi @ gibi yazım kurallarına aykırı olan sayısal ifadeler, noktalama işaretleri ve son olarak da az kullanılan terimler veri setinden ayıklanarak sadeleşme işlemi uygulanmıştır. Veri seti Python'un NLTK kütüphanesinden faydalanılarak belirsizlik içeren “ama, belki, bakınız, eğer, için” gibi kelime çağrışımı net

olmayan ifadelerden arındırılmıştır. Textblob kütüphanesiyle tweetlerdeki metinler tek tek kelimelere ayrıştırılarak metin analizi yapılmıştır. Tek parçaya ayrılan her bir kelime için en çok tekrar eden kelimenin referansı en büyük olacak şekilde referanslar oluşturulmuştur. Bu işlem sonucunda en fazla referans 354 frekans ile teşekkür, 322 frekans ile Sakarya olmuştur. Bu şekilde kelime kullanımı en fazla olan kelimeler bir araya getirilerek görsel bir kelime bulutu oluşturulmuştur. Bu kelime bulutuna bakılarak hizmetlerden memnuniyet hakkında bir fikir edinilebilir. Eğitilen veri sonucunda Sakarya Büyükşehir Belediyesi için olumlu ve olumsuz görüşler birbirine yakın çıkarken diğer ilçe belediyelerinde ise olumlu ifadelerin olumsuz ifadelerle göre daha baskın çıktığı sonucuna varılmıştır. Kategorik olarak bakıldığında ise fen işleri, sosyal hizmetler ve kültürel hizmetlerde en başarılı belediye Serdivan olurken, spor alanında Hendek ve çevre alanında ise Adapazarı'nın daha iyi hizmet verdiği görülmüştür.

Tuzcu (2020) tarafından gerçekleştirilen çalışmada, online bir kitap satış sitesinde 47 farklı kitap hakkında okuyucular tarafından yapılan 91.309 yorum kullanılarak kitap satışlarına ilişkin duygu analizi yapılmıştır. Yorumlar eğitim işlemine geçmeden önce ön işlemeye tabi tutularak yorumlar içerisindeki anlamsız ifadeler, noktalama işaretleri, yabancı kelimeler, sayısal ifadelerden arındırılarak temizlenmiştir. Bu işlem için Python NLTK kütüphanesinden faydalanılmıştır. Eğitime hazır veriler makine öğrenme algoritmalarından MLP, NB, DVM, LR' da işleme sokulmuştur. Eğitime 1400 veri hazırlanmıştır, dağılım %50 olumlu, %50 olumsuz yorumdan oluşmaktadır. Sınıflandırma başarısı ortalama olarak en başarılı algoritma %89 ile MLP, en başarısız algoritması ise %77 ile Naive Bayes olmuştur.

Zhang ve Ananiadou (2022) tarafından yapılan duygu analizi çalışmasında cinsiyetin duygu dağılımı üzerindeki etkisi araştırılmıştır. Bu çalışmanın diğer duygu analizi çalışmalarından birtakım farkları bulunmaktadır. Çünkü cinsiyetten duygu analizi elde edilmesi için veri kümesinin hem cinsiyet bilgileri içermesi hem de duygu etiketleri bulundurması gerekmektedir. Bu yüzden çalışmada hem duygu etiketleri bulunduran hem de cinsiyet bilgilerini içeren iki ayrı veri seti kullanılmıştır. Bunlar Isear ve CrowFlower'dır. Çalışmada veri kümeleri cinsiyetlere göre 'erkeklerin yazdığı cümleler, kadınların yazdığı cümleler ve karma olarak yazılan cümleler' şeklinde üç ayrı gruba ayrılmıştır. Ayrıca veri kümeleri üzerinde eğitilmiş duygu modelleri geliştirilerek duygular farklı kategoriler arasında sınıflandırılarak performans farklılıkları karşılaştırılmıştır. Kullanılan tüm modellerde CNN kullanılmıştır. Çalışma sonuçlarında cinsiyet faktörünün duygu analizi üzerinde haklılığı görülmekle birlikte kadın verilerinin bulunduğu modellerin erkek verilerinin bulunduğu modellere göre daha başarılı olduğu tespit edilmiştir.

Polatgil (2024) tarafından elektrikli araçlar ve yerli otomobil TOGG hakkında Python programlama dili kullanılarak duygu analizi çalışması yapılmıştır. Bu çalışmada elektrikli araçların çevre dostu olması ve artan yakıt fiyatlarından dolayı insanların elektrikli araçlara karşı yaklaşımı ölçülmektedir. Youtube’da TOGG ile ilgili en çok izlenme alan “İlk kez TOGG kullandık”, “Türkiye’nin otomobili TOGG hakkındaki düşünceler” ve “TOGG’u CEO’su ile birlikte kullandık” isimli üç videoya yorum olarak eklenen yaklaşık 30 bin işlenmemiş veri, bu çalışmanın veri setini oluşturmaktadır. Çalışmada Youtube yorumlarının kullanılmasının sebebi ise 2022 TÜİK verilerine göre en çok haberleşme sağlanan sosyal medyanın Youtube olmasıdır. Çalışmada veriler işlenmeden önce zemberek kütüphanesi yardımıyla gereksiz ifadeler ayıklanmış, noktalama işaretleri yok edilmiş ve tüm kelimeler küçük harflere çevrilerek kelimeler kök hâline getirilmiştir. Her bir kelimenin frekansı oluşturularak 3 video için kelime bulutu oluşturulmuştur. Kelime bulutlarına göre en çok tekrar edilen ifadelerin “yerli ve milli”, “Emeği geçenlerden Allah razı olsun” gibi kavramlar olduğu anlaşılmaktadır. Çalışma sonucunda giriş kısmında bahsedilen algoritmalarda eğitilen verilerin yaklaşık %85 oranında olumlu olduğu görülmüş, bu da hem firma açısından hem de TOGG almayı düşünen insanlar açısından bir fikir oluşturmıştır.

Turan ve Arıç (2020) tarafından yapılan çalışmada 61 farklı filminden alınan yaklaşık 50 bin görüntü üzerinden CNN tabanlı derin öğrenme teknikleri kullanılarak duygu analizi işlemi gerçekleştirilmiştir. Çalışma esnasında Amazon web servisi ile videolardan çekilen 50 bin görüntü, işlenmeden önce bazılarının insan yüzünü temsil etmediği veya bozuk olduğu Haarcascade yöntemiyle belirlenmiş ve bu resimler işlem öncesi çıkarılmıştır. Videolardan alınan görüntülerin her birinin isminin ve videonun kaçınıcı saniyesinden alındığı bilgiler bir JSON dosyasında saklanmıştır. 50 bin veriden geriye kalan 20 bin veri ile eğitim işlemi gerçekleştirilmiştir. Bu işlem için “üzgün, sakın, sinirli, mutlu, korkak ve bıkkın” şeklinde yedi ayrı duygu sınıfı oluşturulmuştur. Eldeki verilerin %80’i eğitim verisi için, %20’si ise test için ayrılmıştır. Eğitim sonunda CNN tabanlı modellerden en başarılı modelin %72 ile MLP olduğu sonucuna varılmış, genel olarak ise tüm modellerin ortalama %60 civarında bir başarımla sergilediği görülmüştür. Bıkkınlık ve şaşkınlık duygularının ise en çok birbirine karıştırılan duygular olduğu görülmüştür.

Eyüpınar vd. (2021) tarafından Youtube’da sağlıklı yaşam ve spor ile ilgili en çok izlenme alan altı videodan alınan veriler ışığında duygu analizi çalışması yapılmıştır. Bu çalışmanın önemli kazanımı spora yeni başlayacak olan kişilere fikir vermesi ve beslenme konusunda nelerin faydalı, nelerin zararlı olabileceğine dair ön bilgi oluşturmaktır. Video başlığı olarak "Karbonhidrat ve spor", "Antrenman öncesi beslenme nasıl olmalı?", "BCAA supplement ne

işer yarar?”, “Kas yapmak için ne yapılmalı?” ve “Sporcu market alışverişı” adlı başlıklardan alınan 1902 yorum bu çalışmanın veri setini oluşturmaktadır. Veriler işleme sokulmadan önce Youtube’un kullanıcılara esnek yorum sağlamasından dolayı yorumlardaki noktalama işaretleri, boşluklar, yabancı kelimeler, duygu ifade etmeyen kelime grupları için veri setinden çıkarma işlemi uygulanmıştır. Böylece geriye duygu içeren 1850 ifade kalmıştır. Geri kalan veriler olumlu, olumsuz ve nötr olmak üzere üç ayrı sınıfa ayrılmıştır. Bu çalışmada veri biliminde ve görselleştirmede kullanılan orange3 yazılımı kullanılmıştır. Uygulama sonucunda yorumların %27’si pozitif, %17’si negatif ve %56’sı ise nötr olarak bulunmuştur. Pozitif duygular içerisinde frekansı en yüksek kelime teşekkür, sağlık gibi kelimeler çıkarken negatif duygular içerisinde ise BCAA, kreatin kelimelerin frekansı yüksek bulunmuştur. Nötr yorumlar için ise yumurta, tavuk kelimeleri frekansı ön sıralarda olan kelimelerdir.

Tepecik ve Demir (2024) çalışmalarında Common Voice’tan aldıkları 20.760 ses verisini Naive Bayes, SVM, RF, DT ve KNN makine öğrenme algoritmalarında eğiterek oluşan sonuçları pozitif, negatif ve nötr olarak sınıflandırmışlardır. Bu çalışmada hem Python hem de PapidMiner programlama dili kullanılmıştır. Ses verileri işleme sokulmadan önce ön işlemeden geçirilerek bozuk ve kendini tekrar eden sesler veri setinden çıkarılmış, 5001 veri işleme sokulmak üzere hazır edilmiştir. Veri setindeki verilerin %85’i eğitim, %15’i ise test verisi olarak ayrılmıştır. Verilerin dağılımında 2430 ses nötr, 1989 ses negatif ve 582 ses ise pozitif sınıfa aittir. Bunların 751’i test veri setini oluştururken test datasının içinde 369 nötr, 307 negatif ve 75 pozitif ses bulunmaktadır. Çalışmada %67 ile en başarılı makine modeli NB ve Destek Vektör Makinesi (SVM) bulunmuştur.

Sel (2023) tarafından yapılan çalışmada 2019-2022 yılları arasında tüm dünyayı etkisi altına alan Coronavirüs salgını hakkında detaylı bir araştırma ile insanların duyguları tespit edilmiştir. Bu çalışma, insanların duygularını en çok ifade ettiği Twitter uygulamasında “covid19, Corona, pandemi” gibi konunun içeriğine uygun anahtar kelimelerden oluşan 1,2 milyon veri üzerinden yapılmıştır. Veriler Snsrape metoduyla çekilmiştir. Daha sonra veri, içindeki yabancı unsurlar, noktalama işaretleri ile sayısal ifadelerden temizlenmiştir. Mesajlar Textblob yazılımıyla hangi aya aitse 12 farklı aya ayrılmış ve pozitif, negatif ve nötr olacak şekilde üç ayrı kategoride sınıflandırılmıştır. Çalışma sonuçlarına bakıldığında her aydaki gelişmeler neticesinde insanların virüs hakkında görüşleri saptanmıştır. Örneğin “aşı, solunum, DSÖ, solunum cihazı, hastaneler” gibi kavramların öne çıkması, hastaneler ve DSÖ virüse karşı olumlu bir hava çizerken “salgın, karantina, ölüm, koma, sınav” gibi görüşlerin ön plana çıkması ise olumsuz havanın yüksek olduğunun belirtileridir.

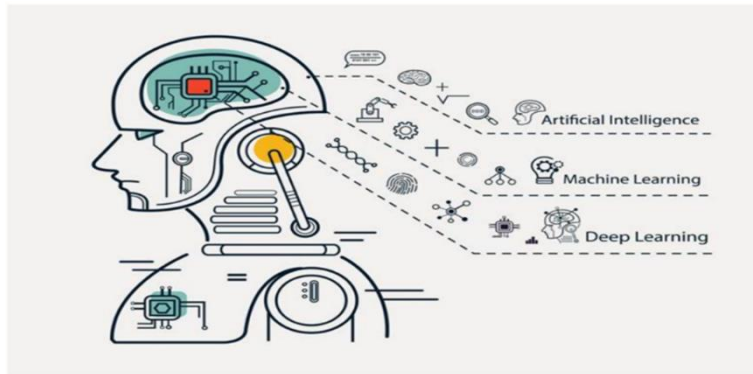
Korkusuz (2019) tarafından futbolseverlerin ve fanatiklerin maçtan sonra Twitter’da yaptığı yorumlar baz alınarak futbol üzerinden duygu analizi çalışması yapılmıştır. Günümüzde kullanıcıların en aktif yorum ve görüş belirttiği platform olan Twitter’dan alınan yorumlardan veri seti oluşturulmuştur. Veriler alınırken Twitter API teknolojisi kullanılmıştır. Alınan veriler modele verilmeden önce bazı ön işleme ve temizleme işlemine tabi tutulmuştur. Metin içerisindeki noktalama işaretleri, hashtag, sayısal ifadeler, yabancı kelimeler, geçersiz ifadeler ve boşluklar giderilmiştir. Bu işlemler için Python’un zemberek kütüphanesi kullanılmıştır. Yazım hataları giderilerek yerlerine yanlış kullanılan kelimelere yakın kelimeler getirilmiştir. Ayrıca bu kütüphane harf silme, ekleme ve değiştirme işlemleri hizmetini de desteklemektedir. Toplam 30.000 veri, el ile etiketleme işlemi yapılarak olumlu, olumsuz, alakasız ve tarafsız olmak üzere dört farklı sınıfa ayrılmıştır. Veriler hazır hâle getirildikten sonra makine öğrenme modellerinden Naive Bayes %84, KNN %87, C4.5 %89, SVM ise %92’lik bir başarımla elde etmiştir. Çizelge 2.1’de bu alanda yapılan çalışmaların literatür tablosu verilmiştir.

Çizelge 2.1. Literatür özeti

Referans	Algoritma/Kütüphaneler	İçerik	Veri kaynağı	Dil	Sonuçlar
Vural vd. (2013)	Dictionary lexicon	Film yorumları	Beyaz perde	Java	-
Bilgin ve Şentürk (2019)	Doc2Vec, Dm, DBow	Sosyal medya yorumları	Twitter	Python	%34 DM kesinlik %44 DBow kesinlik
Demirci vd. (2019)	NN, SVM, LR	Siyasi haberler	Twitter	Python	%81 Doğruluk (ML) %81.86 Doğruluk (DL)
Seyfioğlu ve Demirezen (2017)	Yarı denetimli öğrenme	Uçak Müşteri Yorumları		Python	%92.5 Doğruluk
Zilyas ve Yılmaz (2023)	LR, Rastgele Orman, KNN, Btr	Öğrenci Başarısı Ölçme	Anket sonuçları	Python	%88 Doğruluk
Ayan, Kuyumcu, Ceylan (2019)	Lineer Ridge Regresyonu ve NB	İslamafobi düşünceleri	Twitter'da İslam ile ilgili tweetler	Python	%95.4 F1 skor
Uyaroğlu Akdeniz ve Cebeci (2021)	Turkish Bert	Belediye Hizmeti memnuniyeti	Twitter yorumları	Python	-
Tuzcu (2020)	MLP, NB, LR, DVM	Kitap Satışları	Kitap satış yorumları	Python	%90 DVM doğruluk %88 MLP doğruluk
Yohanes vd. Saputri (2023)	LSTM, BiLSTM, gru	İnsanların temel duygularını araştırmak ve gruplandırmak	ISEAR veri seti	Python	%98.72 doğruluk
Zhang ve Ananiadou (2022)	CNN modeller, Derin öğrenme model	Duygu analizine cinsiyet etkisi	ISEAR ve CrowdFlower	-	%99 doğruluk
Issa vd. (2020).	Derin öğrenme modelleri	Konuşmadan duygu tanımlama	RAVDESS, EMO-DB, IMOCAP	Python	%95.71 EMO-DB doğruluk
Yüksel ve Tan (2018).	NLP, Zemberek, Sade Bayes Algoritması	Foursquare üzerinden gidilen mekanlar hakkındaki duygular	Foursquare alınan yorumlar	Python	%81 ikili sınıflandırma doğruluk %84 üçlü sınıflandırma doğruluk
Ariğ ve Turan (2020)	CNN derin öğrenme modeli	Video duygu analizi	61 film den alınan yorumlar	Python	%85 ML doğruluk
(Eyipınar vd. (2021)	Metin madenciliği Teknikleri	Sağlıklı beslenme ve spor	En çok izlenen 6 spor videosu	Orange 3	%60 doğruluk oranı
Tepecik ve Demir (2024)	Nb, SVM, RF KNN, DT	Sesten duygu analizi	Common Voice	Python ve Rapid Miner	-
Sel (2021)	N-gram yöntemi,	Pandemi hakkındaki görüşler	Twitter	Python	%70 doğruluk
Korkusuz (2019)	NB, KNN, SVM, C4.5	Futbol hakkındaki görüşler	Twitter	Python & C#	-
Parlak (2022)	NB, KA, DVM	Suriyeliler hakkındaki duygular	Twitter	Python	%92.50 doğruluk
Uysal (2024)	Transfer öğrenme modelleri	Duygu analizi	Kaagle	Python	%89 doğruluk

2.2. Yapay Zekâ

Yapay zekâ kavramı son yıllarda hızla yaygınlaşan ve hayatımızın birçok kesiminde kullanılan ve hayatımızı kolaylaştıran bir teknoloji olsa da gerek tanım olarak gerekse kullanım açısından soyut bir olgu olduğundan zihinlerde muğlak bir anlama sahiptir. En açık haliyle yapay zekâ, insanın düşünme yöntemlerini taklit eden yapay yöntemlerdir. Yapay zeka (YZ), bilinmeyen bir durum hakkında eski verileri kullanarak algoritmalar sayesinde yeni bir tahminleme, karar verme yeteneği olarak tanımlanabilir. Bu yüzden insan zekâsı ile karşılaştırıldığında yapay zekânın iyi veya kötü olduğu yorumunda bulunmak mümkün değildir (Karabal, 2023). YZ, insan zekası dâhil olmak üzere doğal zekânın aksine makineler tarafından görüntülenen bir zekâ türüdür. Genellikle akıl yürütme, tahminleme, geçmişe dayalı birikimlere dayanarak öğrenme gibi birçok sistemde kullanılır. Yapay zekânın ortaya çıkışı, kainatın oluşumu ve yaşamın başlangıcından sonraki en büyük üçüncü olay olarak görülmektedir (Pirim, 2006). Çok geniş bir alan olduğu için içerisinde makine öğrenmesi, derin öğrenme, yapay sinir ağları, genetik algoritmalar ve bulanık mantık gibi birçok alt başlık bulunmaktadır. Yapay zekâ 1950’li yıllarda “Makineler insanların davranışını taklit edebilir mi? İnsan gibi kavrayıp düşünebilir mi?” sorularıyla ortaya çıkmıştır. Buna örnek verecek olursak 1997’de IBM’nin Deep Blue adlı satranç oynayan bilgisayarıdır. Deep Blue adlı bilgisayar dünyanın en iyi satranç şampiyonunu yenmiş, insan seviyesi performansının üstüne çıkmıştır. Son 20 yılda ise veri miktarının artması, oyun sektörünün gelişmesi, donanım sistemlerinin iyileştirilmesi ve yeni algoritmaların keşfi ile YZ’nin çağ atlادığı söylenebilir. Bu alanın gelişmesiyle sadece bilişim alanında değil, biyoloji, savunma, nükleer, eğitim gibi birçok alana çözümler sunulmaktadır. Yapay zekânın hayati birçok alt dalı bulunmaktadır. Bunlar makine öğrenmesi, derin öğrenme ve duygu analizi olarak sayılabilir (Varol, 2022). Şekil 2.1’de yapay zekanın alt dalları gösterilmiştir.

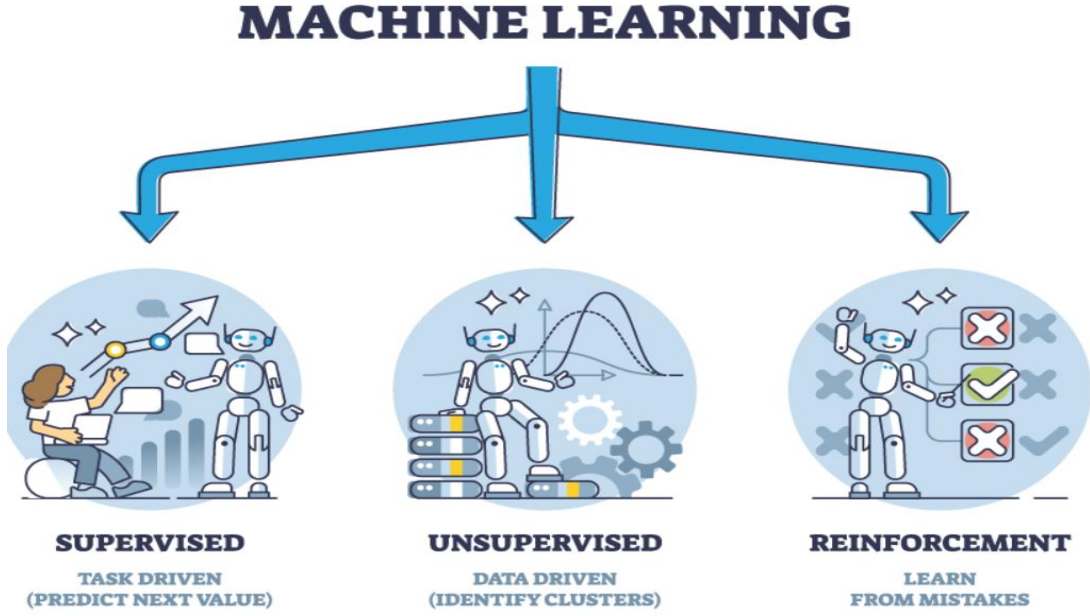


Şekil 2.1. Yapay zekâ ve alt dalları (Çağlar, 2021)

2.3. Makina Öğrenmesi

Veride saklanan bilgileri ortaya çıkarma disiplini olup yapay zekânın bir alt dalıdır. Programlama olmadan sadece tecrübeler yoluyla öğrenmeyi amaçlamaktadır. ML, genellikle yapılan bir analizde sonucu bulmak ve bir sonucu tanımlamak amacıyla kullanılır (Pekel, 2018). Bilgisayarların geçmişteki büyük verileri toplayarak bunları işleyip düzenleme noktasında makine öğrenmesi algoritmaları kullanılır. Eskiden makinelere kurallar izah edilerek bu kurallara göre verilen girdinin sonuçları alınmaktaydı. Makine öğrenmesi ise kuralları makineye vermek yerine verilen girdiye göre makinenin kuralları oluşturması prensibine dayanır. Makine öğrenmesi “Bilgisayarlar insanların yapabileceği işlemlerin ötesine gidebilir mi?” fikrinden doğmuştur.

Makine öğrenmesi her problem için uygun olmayabilir. Nasıl eğitim gerçekleştirdiğimiz, makineye sunulan veriler, kullanılan etiketler ve veri kalitesi algoritmanın kalitesini değiştiren ölçekler olmaktadır. Makine öğrenmesinde bayes teoremi, standart sapma, korelasyon gibi çok sayıda temel matematiksel işlem kullanılır. Etkili bir makine öğrenmesinde öznitelik çıkarımı son derece önemlidir. Örneğin makine öğrenmesinin kadın-erkek resminin ayırt edilebilmesi için bir insan gibi özniteliklerin içselleştirilmesi şarttır. Görüntü işlemeye de hitap eden bu örnekte kadın veya erkeğin sahip olduğu kaş yapısı, çene yapısı, saç şekli, çene uzunluğu ve görüntüsü gibi niteliklerin makineye iyi bir şekilde aktarılması önemlidir. Fakat yüz tanıma problemlerinde hâlâ istenilen başarımın elde edilememesinin birçok sebebi bulunmaktadır. Söz gelimi cinsiyet ayrımında herkes farklı bir bakış açısına sahip olabilir. Bu farklılıklar genel ortalama görüşe uygun olmadığı takdirde algoritma doğru sonuca uzak sonuçlar verebilecektir. Verilen örnek basit bir cinsiyet ayrımı olup veri karmaşık ise veya görüntü, ses, video gibi nesnelerden oluşuyorsa makine öğrenme algoritmaları bu problem için iyi bir seçim olmayacaktır. ML temelde üç aşamaya ayrılır: denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme (Zilyas ve Yılmaz, 2023). Şekil 2.2’de makine öğrenmesinin çeşitleri verilmiştir.



Şekil 2.2. Makine öğrenmesi çeşitleri (Cankut, 2023)

2.3.1. Denetimli Öğrenme

Denetimli öğrenme, makine öğrenmesinde en çok kullanılan yöntemlerinden biridir. Veri setinin sahip olduğu bilgileri kullanarak öğrenir. Kullandığı bu bilgileri yeni bir sorgu veya problemle karşılaştığında geçmişteki tecrübelerden faydalanarak problemi çözer. Çıktı olarak sayısal bir değere dönüştürebildiği gibi kategorik bir sonuç da üretebilir. Tahmin edilen sonuçların performans değeri tahmin edilen sonuç kadar değerlidir. Çoğunlukla tahmin edilen sonuç ile birlikte bu sonucun kalitesine yönelik güven aralığı, olasılık ve doğruluk değerleri de belirlenir. Denetimli öğrenme ile çalışan iki alan vardır: sınıflandırma ve regresyon. Denetimli öğrenmede etiketleme işlemi uygulanmaktadır. Model, veri kümesini anlamak için etiketli verileri kullanarak modeli inşa eder ve eğitim ve test işlemi tamamlandıktan sonra çıktının doğruluğunu kontrol etmek için örnek bir veri modelde test edilir. Örneğin nesne tespiti yapan bir model için binlerce araba resmi araba veya taşıt etiketiyle etiketlenmelidir. Bu işlem sonunda makine öğrenme modeline bir araba resmi verildiğinde eğitim verisinden faydalanarak model, bunun bir araba resmi olduğunu anlayacaktır. Kısacası denetimli öğrenmede temel amaç girdi ve çıktı verileri arasındaki eşleşmenin gerçekleşmesidir (El Bekri, 2020).

2.3.2. Denetimsiz Öğrenme

Denetimsiz öğrenme makine öğrenmesinin bir diğer alt dalı olup kümeleme ve ilişkilendirme problemlerinde uygulanır. İlişkilendirme algoritması veriler arasındaki

bağlantıdan bulunurken kümelenendirme işlemi ise verilerin yorumlanmasıyla elde edilir (Candan, 2021).

Denetimsiz öğrenme, herhangi bir bilgiye sahip olunmadan işlem yapma pratiğidir. Denetimli öğrenmede olduğu gibi veriler etiketli değildir. Burada model kendi öğrenerek veriler eğitilmektedir. Denetimsiz öğrenmenin birçok avantajı bulunmaktadır. Etiketleme işleminin bulunmaması zamansal açısından bir tasarruftur. Ayrıca işlenmemiş verilerin anlaşılmasına yardımcı olacağından veri uzmanları açısından kullanışlı bir yöntemdir. Bu yöntemde verilerin ne kadar eksik olduğu da bulunabilmektedir. Yavaş bir öğrenme gerçekleştirdiğinden insan zekâsına benzetilebilir (Hazır, 2021).

Bunların yanı sıra öncesinde modele bir bilgi verilmediğinden dolayı yöntemin doğruluk oranı düşüktür. Algoritmanın tüm olasılıkları adım adım hesaplamasının vakit alması, tercih edilmeyen yönüdür.

2.3.3. Pekiştirmeli Öğrenme

Yazılımın en iyi sonuçları elde etmek üzere kararlar vermesi için eğitildiği bir makine öğrenmesidir. Bu yöntem, insanların hedeflerine ulaşmak için kullandığı deneme yanılma yoluyla öğrenme sürecini tercih eder. Pekiştirmeli öğrenme önceden etiketlenmiş verilere dayalı olmayıp ancak etkileşim ve denemeler neticesinde öğrenme sağlayabilir. Bu algoritma ödül-ceza mekanizması üzerine çalışır. Algoritma bir görevi başarıyla görevi yerine getirdiğinde ödül alırken hatalı eylemlerinde ise ceza almaktadır. Pekiştirmeli öğrenme algoritmaları gayet esnektir.

Pekiştirmeli öğrenme algoritmaları belirsiz ortamlarda bile uyumlu bir şekilde çalışabilir. Buna rağmen adaptasyon problemi sorunu yaşamaz. Sürekli gelişim sağlar. Deneyimlere dayanarak öğrenme sağlar. Bu yüzden sürekli gelişmekte ve değişmektedir (Çağlar, 2021).

2.4. Yapay Sinir Ağları (YSA)

Derin öğrenme insan beyninden esinlenerek öğrenen bir yapay zekâdır. Derin öğrenme daha kompleks olan resim, video, ses gibi modelleri tanıyabilir. Görüntü analizi, sesi metne çevirme, metinden veya sestten duygu analizi gerçekleştirme gibi zor problemlerde bu sistemler kullanılır. Bu sistemler, öğrenme yoluyla bilgiyi üretme ve keşfetme yeteneklerini bir yardım almadan gerçekleştirmeyi amaçlar. Derin öğrenmede verilerin öğrenmesi yapay sinir ağlarında gerçekleşmektedir. İlk sinir ağı modeli 1943 yılında bir sinir hekimi olan Warren McCulloch ve Walter Pitts tarafından ortaya çıkarılmıştır (Öztürk ve Şahin, 2018). Yapay sinir ağındaki her bir bileşen insan sinir sistemindeki bir parçadan esinlenmiştir. Örneğin nöronlar işletim

elemanını, dentritler toplama fonksiyonunu, hücre gövdesi transfer fonksiyonunu, aksonlar yapay nöron akışını temsil ederken sinapslar ise ağırlıklara karşılık gelmektedir. Derin öğrenmenin en zor yanları eğitim için çok büyük veri kümelerine sahip olunması ve çok fazla hesaplamaya ihtiyaç duyulmasıdır (Day ve Khoshgoftaar, 2017; “Python Programlama Dili Nedir?”, 2021).

YSA, makine öğrenmesinin bir alt dalıdır. Makine öğrenmesi algoritmaların ve istatistiksel modellerin bilgisayar etkisinden uzak bir şekilde tahmin etmesi iken yapay sinir ağları ise insan beynini taklit eden bir makine öğrenmesidir. Veriler arasındaki ilişkiyi öğrenerek yüksek performanslı çıktılar verebilir. Yapay sinir ağı filtreleme, veri ilişkilendirme, sınıflandırma gibi birçok alanda kullanılabilir. Yapay sinir ağı, problemin türüne göre avantaj ve dezavantaj gösterebilir. YSA’nın avantajlarından bazıları şunlardır:

- Eldeki bilgilerin tamamı veri tabanı yerine ağlarda saklanabilir.
- Eğitilen yapay sinir ağları eğitim sonrasında eksik veriler olsa bile çıktı verebilir.
- Hataya karşı tolerans değerleri bulunmaktadır.
- Birkaç hücrenin bozuk veya hatalı olması çıktıyı etkilemez.
- YSA birçok hücreden meydana gelmesine rağmen aynı anda birden çok işlemi gerçekleştirebilir.
- Denetimli ve denetimsiz öğrenme seçenekleri vardır.

Yapay sinir ağlarının temel birimi nöronlardır. Nöronlar sinir hücrelerine benzer şekilde bilgi işleme yeteneğine sahiptir. Bir nöron girdi verilerini alır, bu girdileri işler ve sonuç olarak bir çıktı üretir. Yapay sinir ağının en küçük parçası olarak bilinen algılayıcı, aşağıdaki gibi bir lineer fonksiyon ile ifade edilmektedir: $y=Wx$

Burada;

x: Bağımsız değişken veya girdi olarak adlandırılır. Bir problemin girdilerini temsil etmektedir.

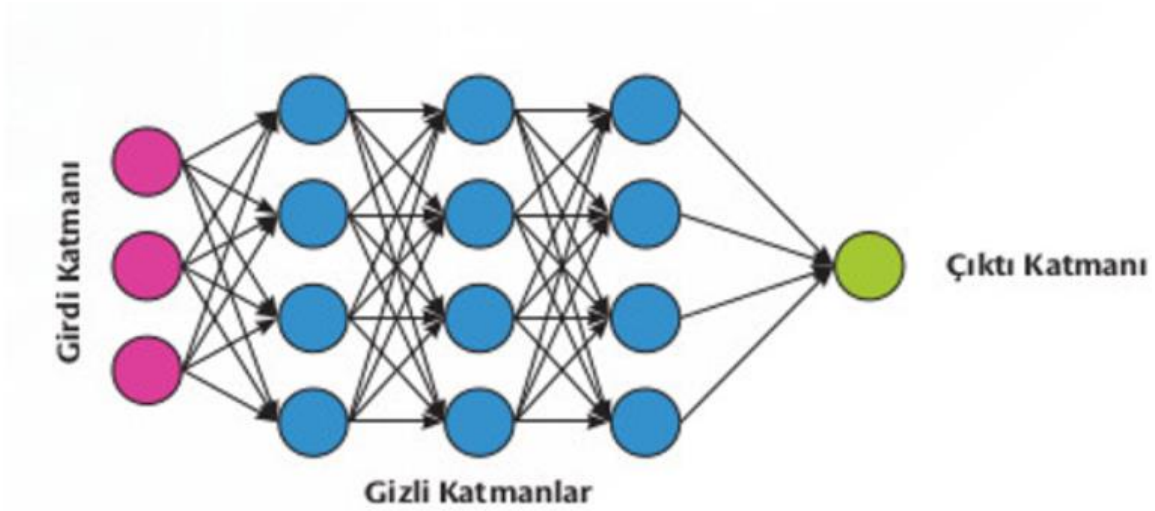
y: Bağımsız değişkenine bağlı olarak hesaplanan ve sonucu temsil eden bağımlı değişkendir.

W: Ağırlık parametresi, girdi verileri (x) ile çarpıldığında, çıktıyı (y) etkileyen faktörlerdir.

Yapay sinir ağı birden çok nöronun bir araya gelmesiyle oluşurken yapısı bir giriş katmanı, bir çıkış katmanı, giriş ve çıkış katmanı arasında bir veya birden fazla gizli katman barındırabilir. Bir katmandaki her nöron bir üst katmandaki her bir veri ağırlık değerleriyle çarpılarak gizli katmana iletilir. Sonuç, gizli katmanda bir transfer fonksiyondan geçirilir ve

çıktı katmanına gönderilir. Gizli katmanda 0 ile 1 arasında değişen çıktı elde etmek için transfer fonksiyonu olarak doğrusal sigmoid fonksiyonu kullanılabilir. YSA doğrusal değildir.

Doğrusal modeller önemli detayları yakalayabildiği takdirde avantajlıdır. Şekil 2.3'te Yapay sinir ağlarının içerdiği katmanlar gösterilmiştir.



Şekil 2.3. Yapay sinir ağı katmanları (Öztürk, 2022).

2.4.1. Yapay Sinir Ağı Katmanları

2.4.1.1. Giriş Katmanı

Bu katmanda herhangi bir bilginin işlenmesi söz konusu değildir. Dış evrendeki verilerin ağa alındığı katmandır.

2.4.1.2. Gizli Katman

Gizli katmanlar verinin özelliklerini temsil etmektedir. Bu katmanlar girdi verilerini daha yüksek seviyeli özelliklere dönüştürerek veri setindeki karmaşıklığı yakalamaktadır. Gizli katman problemin çeşitliliğine göre birden fazla olabilmektedir. İlk gizli katmanda temel özellikler yakalanırken sonraki her bir katmanda bu özellikler daha karmaşık özelliklere dönüşmektedir. Resmin girdisinin kenar, renk ve desen gibi özellikler gizli katmanda çıkarılır. Yine gizli katmanlar sayesinde aşırı öğrenme engellenir. Aşırı öğrenmenin fazla olması durumunda model kötü bir öğrenme performansı sergiler. Gizli katmana gelen veriler girdi katmanı ile gizli katman arasında bulunan bağlantı katmanları ile çarpılıp gizli katmana iletilir. Burada gelen veriler aynı şekilde gizli katmanı ile çıktı katmanı arasındaki bağlantı ağırlıkları ile çarpılıp çıktı katmanına gönderilmektedir (Asilkan ve Irmak, 2009).

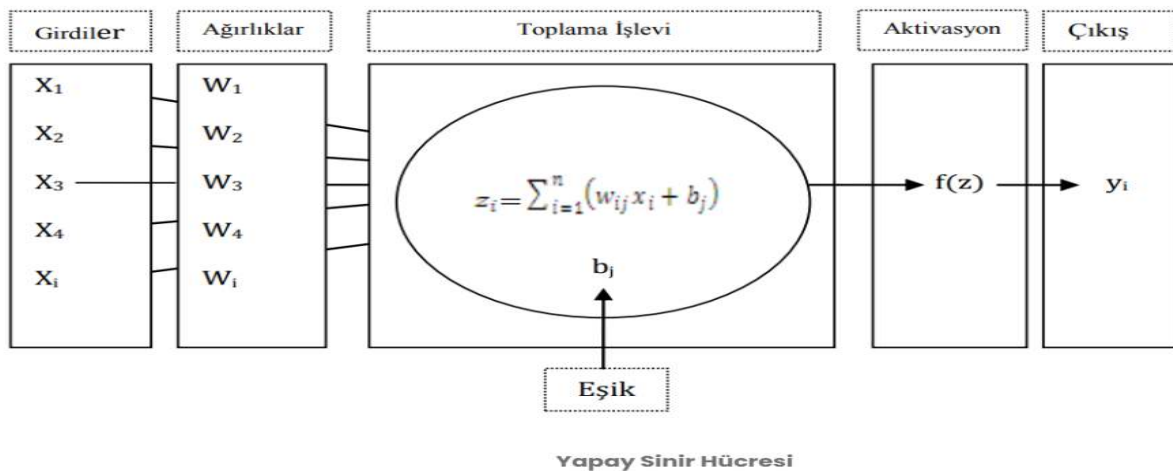
2.4.1.3. Çıktı Katmanı

Sistemin girdi verilerine dayanarak öğrenmesi istenen çıkış değerini üreten katmandır. Tahmin ve sonuçların elde edildiği yerdir. Eğer sorun sınıflandırma problemi ise çıkış katmanı etiket çıktıları verirken regresyon problemi sayısal sonuç üretir. Aktivasyon fonksiyonu ise nörondan gelen bilginin bir sonraki nörona iletilip ileilmeyeceğine karar veren birimdir.

2.4.2. Yapay Sinir Ağlarının Temel Yapısı

Bir yapay sinir ağının temel yapısı 5 ayrı parçadan oluşmaktadır.

- *Girdiler*: Girdiler nörondan gelen verilerdir. İnsandaki sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir.
- *Ağırlıklar(weights)*: Girdilerle ilişkilendirilmekte olup çıktı belirtisi için kullanılır. Gelen bilgiler çekirdeğe varmadan önce geldikleri bağlantılarla ağırlıkları çarpılarak çekirdeğe gönderilir.
- *Toplama Fonksiyonu*: Toplama işleminden sonra elde edilen sonucun nöron çıktısının belirlenmesini sağlar.
- *Aktivasyon Fonksiyonu*: Önceki katmandaki girdilerin ağırlıklarını alan ve daha sonra doğrusal olmayan bir çıkış üreten ve çıktıları sonraki katmana geçiren fonksiyondur. (Relu, Sigmoid gibi)
- *Çıktı*: Aktivasyon fonksiyonu sonucunda elde edilen değerlerdir. Bu çıktı sonraki katmandaki nöronlara iletilir ve sonuç olarak kullanılır. Şekil 2.4'te bir yapay sinir hücresini oluşturan bileşenler gösterilmiştir.



Şekil 2.4. Sinir hücresinin bileşenleri (Data, 2021).

2.4.3. Aktivasyon Fonksiyonları

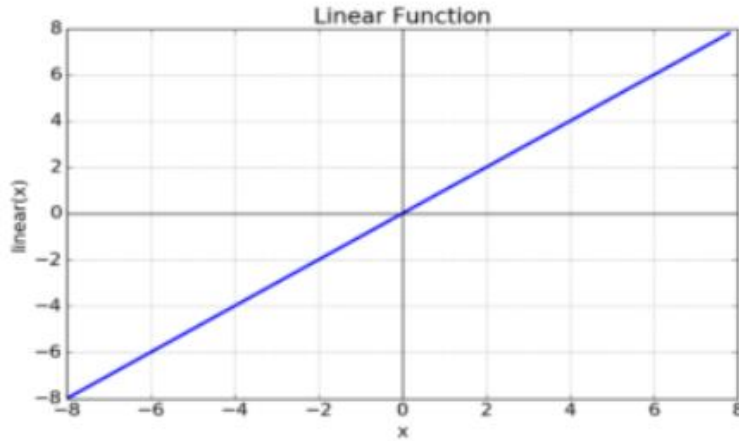
Bu fonksiyonların işlevi toplam fonksiyondaki çıktıların nasıl bir değişim içinde olduğunu belirlemektir. Bu fonksiyonlar sayesinde modellerin karmaşıklığı artırılmaktadır, bu sebeple YSA modelleri veri setindeki en karmaşık yapıları bile öğrenerek uygulandığı problemlerde iyi sonuç vermektedir. Aktivasyon fonksiyonları doğrusal değildir. Nöron çıktıları belli bir değer arasında sıkıştırılır. Bu değerler $[-1,1]$ veya $[0,1]$ olabilmektedir. Çıktı değerleri -1 ve 1'e ne kadar yakınsa tahminleme doğrularına katkısı o kadar yüksektir. 0'a yakın olduğunda ise tahmin değerine etkisi azdır. İyi bir sonuç alınması için aktivasyon fonksiyonunun seçimi de önemli olmaktadır ("Aktivasyon Fonksiyonları Nedir?", 2021)

Aktivasyon fonksiyonu bazı özelliklere sahiptir. Bunlar;

- Aktivasyon fonksiyonu doğrusal olmamalıdır.
- Sürekli olmalıdır.
- Değeri belirli bir aralıkta olmalıdır. Sonsuz çözümde olmamalıdır.
- Monoton olmalıdır. Yani yönü değişmemelidir.

2.4.3.1. Çıktı katmanında kullanılan aktivasyon fonksiyonları

Doğrusal (lineer fonksiyon): Regresyon problemlerinde kullanılır. Girdi değerlerini değiştirmeden çıktı olarak vermektedir. Doğrusal fonksiyon, regresyon problemlerinde çıktı katmanında kullanılabilir. Şekil 2.5'te doğrusal fonksiyon grafiği gösterilmiştir.



Şekil 2.5. Doğrusal fonksiyon grafiği (Ray, 2019)

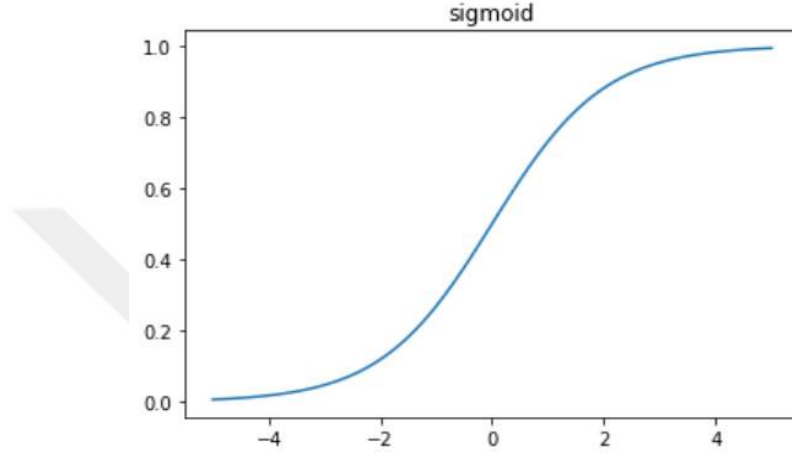
Sigmoid fonksiyonu: Tekli sınıflandırmada kullanılır. Gelen veriler $[0,1]$ arasında değer alır. Lojistik fonksiyon olarak da bilinir.

$$f(x) = \frac{1}{1+e^{-x}} \text{ şeklinde formüle edilir.}$$

x: girdi değeri

e: eular sabitidir. (e: 2.71)

Sigmoid fonksiyonunda eğim girdi değerlerine göre değişebilmektedir. Bu da öğrenme sürecine doğrudan etki eden unsur olmaktadır. Eğer eğim 0'a yakınsa fonksiyon eğimi çok düşer. Bu durum öğrenme hızını yavaşlatıp durdurabilir. Ağın performansını olumsuz etkiler. Buna *vanishing gradient* denir. Çözüm ise ReLU fonksiyonu kullanmak veya normalleştirme işlemi yapmaktır. Şekil 2.6'da sigmoid fonksiyon grafiği gösterilmiştir.



Şekil 2.6. Sigmoid fonksiyon gösterimi (Öğrekçi, 2020)

Softmax Fonksiyonu: Çoklu sınıflandırma problemlerinde kullanılır. Aldığı değer aralığı [0,1] arasındadır. Birden fazla boyutla çalışır.

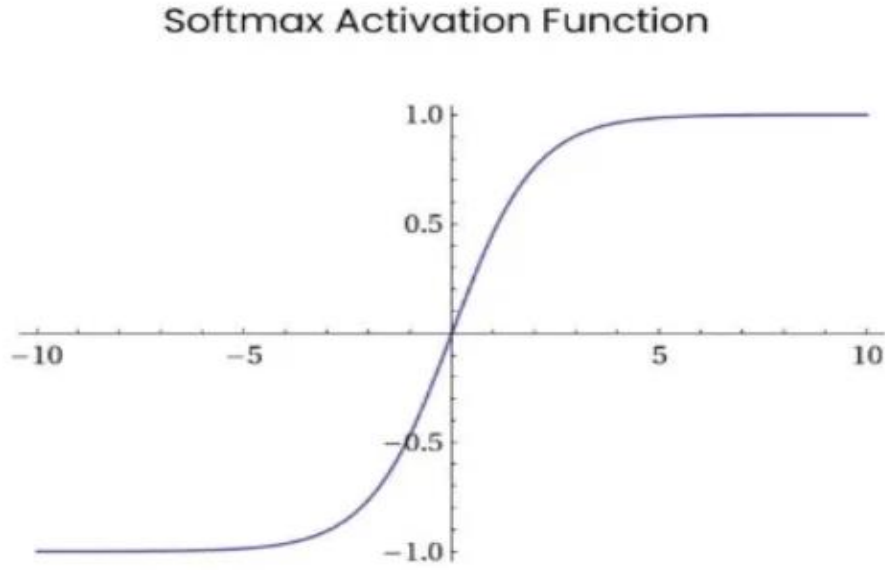
$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (2.1)$$

x_i : girdi vektörünün i. elemanı

K: Sınıf sayısı

E: euler sabiti (e=2.71)

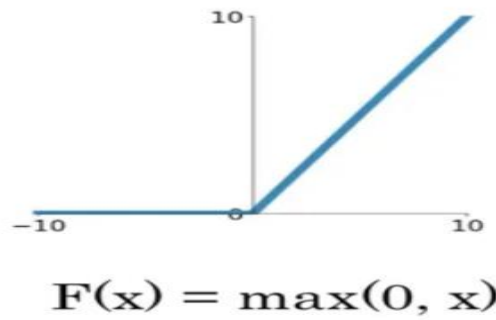
Softmax fonksiyonu her girdi değerini pozitif olarak yorumlar ve normalleştirir. Softmax, VGG16 bilgisayarla görme problemlerinde kullanılan popüler bir modeldir (Çolmabey, 2021). Şekil 2.7'de Softmax grafiği gösterilmiştir.



Şekil 2.7. Softmax fonksiyon grafiği (Yılmaz, 2020)

2.4.3.2. Gizli katmanda kullanılan aktivasyon fonksiyonları

ReLU fonksiyonu: Bu fonksiyon negatif girdileri sıfır olarak atamaktadır. Pozitif değerleri ise olduğu gibi alır. Tercih edilme sebebi hızlı olmasıdır. Fonksiyonun değer aralığı $[0+\infty)$ arasındadır. Bu fonksiyonun diğerlerinden farkı, hesaplama yükünün diğerlerinden az olması ve çok karmaşık problemlerinde kullanılmasıdır. ReLU fonksiyonunda 0'dan büyük olan girdi değerlerin türevi 0'dır. Bu da ağı hızlı eğitilmesini sağlar (Akbaş, 2021). Şekil 2.8'de ReLU fonksiyon grafiği verilmiştir.



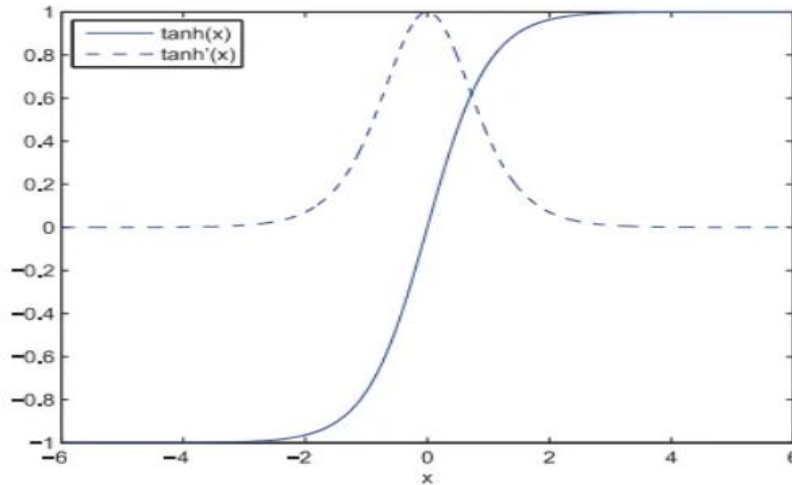
Şekil 2.8. ReLU fonksiyon grafiği

Tanh (hiperbolik Tanjant) fonksiyonu: Bu fonksiyon sigmoid'e benzemesine karşın türevi alındığında daha çok değer alabilmektedir. Bu da daha hızlı öğrenmeyi sağlamaktadır.

Fakat fonksiyonun uç noktalarında yine aynı sorun ile karşılaşilmektedir. Bu fonksiyonun değer aralığı $[-1,1]$ arasındadır. Formülü şu şekilde hesaplanır:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

Şekil 2.9’da Tanh fonksiyon grafiği verilmiştir.



Şekil 2.9. Tanh fonksiyon grafiği (Ayten, 2021).

2.5. Derin Öğrenme

Derin öğrenme, makine öğrenmesinin bir alt dalı olup, çalışma mekanizması olarak insan beynini ve nöronları baz alan bir öğrenme şeklidir. Ham veriden yüksek seviyeli özellikler çıkarmak için yapay sinir ağları kullanılır. Derin öğrenme modelleri birden fazla veri kaynağından bilgileri alır ve bu verileri insana ihtiyaç duymadan çözümler. Derin öğrenmede, grafik işleme birimi aynı anda birden fazla hesaplamayı işleyebildiğinden eğitim modellerine yönelik optimize edilmiştir. Derin öğrenme, çok katmanlı sinir ağı olarak bilinir. Normal sinir ağları bir veya iki gizli katmana sahipken derin öğrenme yüzlerce katmana sahiptir. Bu katmanlar sayesinde verinin özniteliklerine erişilir ve doğruluk değeri yüksek bulunabilmektedir. Derin öğrenme bugün otonom araçlarda, yüz tanımda, müşteri hizmetlerinde, hastalık teşhisinde ve oyun sektöründe sıkça tercih edilmektedir. Derin öğrenmenin gelişimine yol açan bazı önemli faktörler bulunmaktadır.

YSA çok eski tarihlere dayanmaktadır. Yapay sinir ağları ilk olarak 1943 yılında tanıtılmıştır. YSA’nın ilk teorik çalışma ve uygulaması 1960’lı yıllarda geliştirilmiştir. Son yıllarda derin öğrenmeyle ilişkili yapay sinir ağları kullanılarak birçok yapay zekâ problemi çözülmüştür. Derin öğrenme milyonlarca veya milyarlarca özellik içeren matematiksel işleme

uğraşır. 1990'lı yıllarda CPU'lar çok güçlü olmasına karşın bu işlemleri yapamıyordu. Dünyada internetin gelişmesiyle veri miktarı artmış ve veriye ulaşmak kolaylaşmıştır. Bugün kullanımı en popüler arama motoru olan Google ve en yaygın sosyal platformlar sayesinde büyük miktarda veri toplanmaktadır. Toplanan bu veriye büyük veri denmektedir. Bugün derin öğrenme ile problem çözmek için kullanılabilecek verinin çoğuna ücretli veya ücretsiz olarak kolayca ulaşılabilir. Derin öğrenmenin gelişmesini sağlayan diğer önemli faktör ise Python programlama dilinin yaygınlaşmasıdır. Eskiden bu işlemler için C++ kullanılmaktaydı ve GPU ile çalışmak için grafik kartlarının mimarisini bilmek gerekiyordu. Python ile sinir ağlarını programlamak ve derin öğrenme tekniklerini uygulamak daha kolay gelmiştir. Artık programcıların CPU'nun teknik özelliklerini bilmelerine ve mimarilerine odaklanmalarına gerek kalmamıştır. Son birkaç yıldır birçok organizasyon derin öğrenme tekniklerini uygulamak ve daha da basitleştirmek için Python platformları geliştirmişlerdir. Bu platformlar farklı mekanizmalar olmasına rağmen hemen hemen aynı operatörleri kullanmaktadır. Bu platformların çoğu ücretsiz olup açık kaynaklıdır. Derin öğrenme modelleri birden fazla veri kaynağından bilgi alıp bu bilgileri analiz etmektedir. Eğitim sırasında GPU'lar aynı anda birden fazla hesaplamayı işleyebildiği için optimize edilmiştir.

2.5.1. Derin Öğrenmede Kullanılan YSA Modelleri

Derin öğrenme modellerinde problemin türüne göre kullanılmak üzere yaygın olarak kullanılan dört farklı yapay sinir ağı bulunmaktadır.

2.5.1.1. Recurrent neural networks (RNN)

Tekrarlanan sinir ağı olarak bilinir. Sıralı verileri işlemek için kullanılır. Bir sonraki adımı tahmin etmek için kullanılan YSA'dır. Diğer sinir ağlardan farkı, girdi değerlerinin birbiriyle ilişkili olmasıdır. Yani hatırlatıcı özelliği en büyük farkıdır. Bu sinir ağı geçmişteki verilerin birbiri ile ilişkisine dayanarak bir sonraki adımı tahmin etmektedir (Akca, 2020). RNN'de 2 RNN katmanı ve 2 yoğun katman bulunur. Bu katmanlarda sırasıyla 64 ve 32 nöron bulunmaktadır. Ayrıca bu yoğun katmanda ReLU aktivasyon fonksiyonu tercih edilir. Çıkış katmanı için 3 nöron bulunduran RNN sinir ağı bu katmanda softmax fonksiyonu kullanılır (Akca, 2020).

RNN'de geriye doğru besleme (backpropagation) algoritması kullanılır. Bu algoritmanın faydası hatanın kaynağının bulunması ve ağ parametrelerinin optimal seviyede tutulması için önemlidir. Kısacası bu algoritma hata azaltılması için gereklidir. RNN doğal dil işlemede, dizisel veri işlemede, konuşma tanıma işlemlerinde kullanılmaktadır.

RNN'nin temel formülü şu şekildedir:

$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{hx} \cdot x_t)$ şeklinde ifade edilir. Burada;

(h_t) : t zamanındaki gizli katman çıktısı

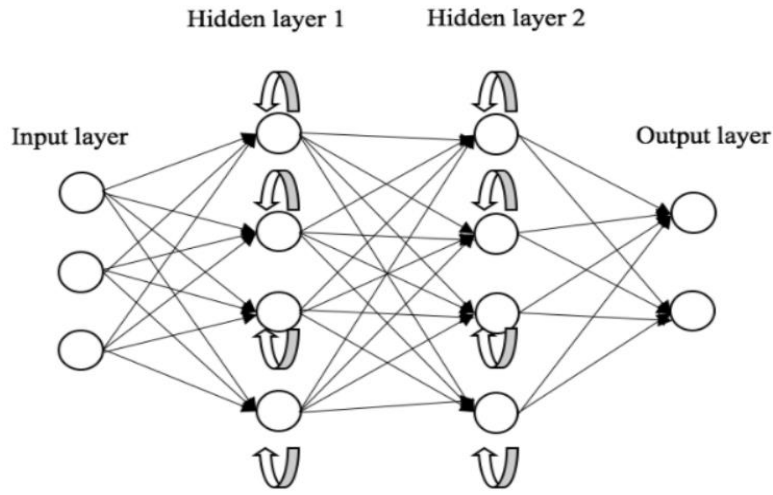
(h_{t-1}) : t-1 zamanındaki gizli katman çıktısı (önceki adımdan gelen bilgi)

(x_t) : t zamanındaki girdi (veri noktası)

(W_{hh}) : Bir önceki gizli katmanın ağırlığı

(W_{hx}) : Şu anki gizli katmanın ağırlığı

$(\tanh)$: Aktivasyon fonksiyonu (genellikle hiperbolik tanjant olarak bilinir). Bu formülle gizli katman çıktıları güncellenir ve sonuçları bulunur. Şekil 2.10'da RNN yapısı ve katmanları gösterilmiştir.



Şekil 2.10. RNN katman yapısı (Community, 2020).

2.5.1.2. İleri beslemeli ağ

En basit sinir ağıdır. Dışarıdan gelen bilgiler giriş katmanında herhangi bir değişikliğe uğramadan sonraki katmana gönderilir. Düğümler hiçbir zaman bir döngü oluşturmaz. Bir giriş katmanı, bir gizli katman ve bir çıkış katmanına sahiptir. Bu tür ağlarda, bilgi sadece giriş katmanından gizli katmanlar aracılığıyla çıkış katmanına doğru ilerler, asla geriye döngüsel bir iletim olmaz. İleri beslemeli ağlar çok katmanlı algılayıcılar olarak da adlandırılır. Doğrusal olmayan problemlerde kullanılmaktadır (Johnson, 2024).

2.5.1.3. Long-short term memory (LSTM/uzun kısa vadeli hafıza ağları)

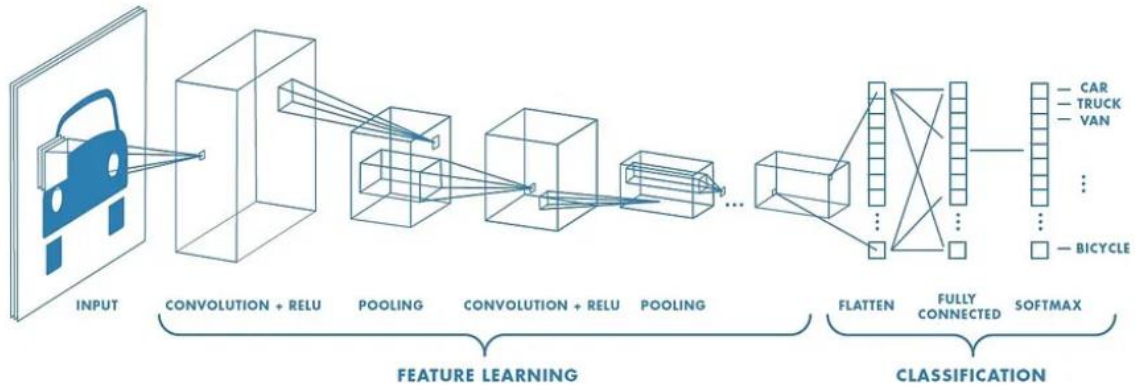
Derin öğrenmede kullanılan bu sinir ağı verileri sırası ile işler ve zaman içinde gizli durumunu koruyabilir. LSTM, RNN'in kısa vadede bellek sorununu çözmek için tasarlanmıştır. Bu ağ içerisinde tekrar eden aktivasyon işlemi uygulanmaz. Gradyan kaybolma

problemi yaşanmaz. Yine farklı olarak RNN tek bir tanh içerirken LSTM’de 4 farklı katman vardır (Doğan, 2020).

RNN kısa vadeli hafızaya sahip olmadığından iyi bir çalışma gösteremez. Geçmiş bilgileri hatırlayamaz. Gradyan değerinin de küçülmesinden dolayı RNN’de öğrenme problemleri ortaya çıkmaya başlar. Bu yüzden bilgileri daha iyi depolamak için LSTM yapay sinir ağları kullanılmaya başlanmıştır (Akköse, 2021). LSTM modeli 2 LSTM katmanı ve 2 yoğun katmandan meydana gelmektedir. Bu katmanlarda 128 ve 64 nöron bulunur. Yine çıktı katmanı diğer modellerde olduğu gibi 3 çıkıştan oluşur (Ezgi ve İhsan, 2022).

2.5.1.4. Evrişimli sinir ağları (CNN)

Genellikle görüntü işlemede kullanılan girdi olarak resim veya görüntüleri baz alan bir yapay sinir ağıdır. Diğer sinir ağı gibi ağırlık ve sinirlerden oluşmaktadır. Şekil 2.11’de CNN yapısının ve katman yapısı gösterilmiştir.



Şekil 2.11. CNN yapısı (Torres, 2023)

ESA modelleri görüntü ve videolardan obje tanıma, sınıflandırma, hastalık teşhisi gibi problemlerde kullanılır.

CNN modelin katmanları şu şekildedir:

1. *Evrişim Katmanı (Convolutional Layer)*: CNN modelin ilk katmanıdır. Giriş verilerini, filtre ve bir öznitelik haritasını içerir. Resimlerin özellikleri tespit edilir. Görüntüdeki özellikleri filtreler aracılığıyla çıkarılır. Bu filtreler çok boyutlu olup genellikle (5x5x3) boyutundadır. Yani 5 yükseklik ve genişlikte 3 derinliğine sahiptir. Evrişim katmanında matrisler üzerindeki işlemler aşağıdaki şekilde gerçekleşmektedir. Şekil 2.12’de CNN mimarisindeki resim ve filtrelerin matris boyutlarının temsiline yer verilmiştir.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Resmin matrisleri

1	0	1
0	1	0
1	0	1

Filtrenin matrisleri

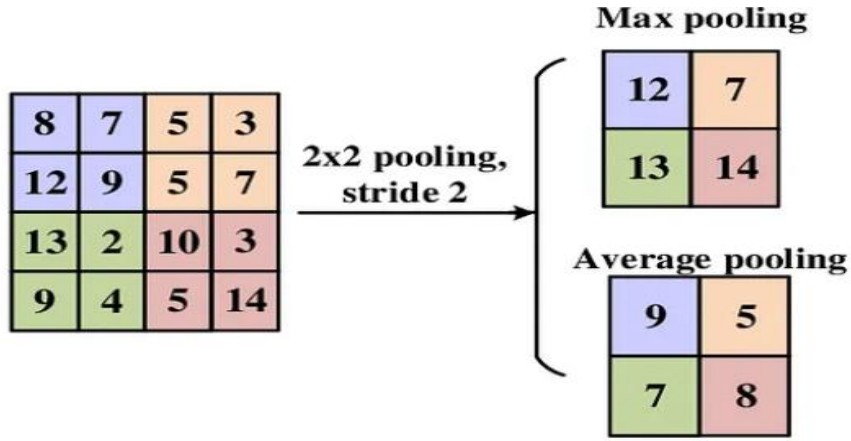
4	3	4
2	4	3
2	3	4

Çıktı matrisi

Şekil 2.12. Evrişim katmanında matris işlemi (Ergin, 2020).

Yukarıda belirtilen 1. matris tablosu resimden alınan özelliklerin sayısal değerlerini ifade etmekte olup 5x5 boyutundadır. 2. matris tablosu ise özellik çıkarımı için gerekli olacak olan 3x3'lük bir filtre matrisidir. 3. matris tablosu ise bulunan çıktıların oluşturduğu matristir. Bu işlemler yapılırken önce belirlenen filtre resmin sol üst köşesinden başlanarak 2 matris değerlerinin (resim ve filtre) birbirine karşılık değerleri çarpılır. Çıkan sonuçlar toplanır ve çıktı matrisine yerleştirilir. Ardından bu filtre her defasında bir sağa kaydırılarak işlem tekrar ettirilir. Buna stride işlemi denir. Fotoğraf üzerinde kaç piksel kayacağı belirlenir. Birinci satırdan sonra ikinci satıra ardından işlemler üçüncü satıra kaydırılarak devam ettirilir. Bulunan çıktıların *Özellik Haritası* denir. Filtreler genelde rastgele başlatılır ve ardından bir optimizasyon süreci kullanılarak eğitilir. Bu süreç esnasında, filtreler görüntü üzerinde kaydırılır ve her konumda bir çıktı değeri hesaplanır. Bunun için CNN'de genellikle birden fazla konvolusyonel katman bulunur. Öznitelik haritası çıkarıldıktan sonra stride (adım) yöntemi ile görüntünün filtre etkisinde kalıp kalmadığı kontrol edilir. Çıkan görüntü orijinal görüntüden daha küçük boyutta olacağı için boş olan yerlere 0 atanır. Bu işleme doldurma(padding) denir.

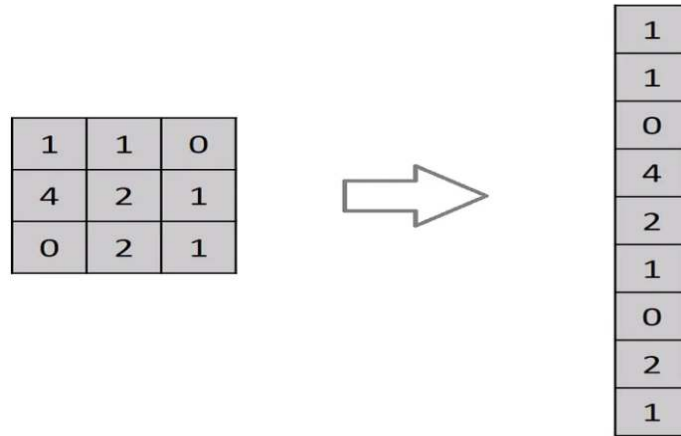
2. Havuzlama Katmanı (Pooling Layer): Bu parametre isteğe bağlı olup her problemde kullanılmaz. Hesaplama işlemini kısaltmak ve parametreleri ayarlamak için kullanılır. Boyut azalma işlemini sağlamaktadır. Minimum, maksimum ve ortalama olmak üzere 3 tür havuzlama vardır. Şekil 2.13'te havuzlama katmanındaki işlemlerin gösterimine yer verilmiştir.



Şekil 2.13. Havuzlama katmanının gösterimi (Torres, 2023).

Havuzlama katmanında maksimum havuzlama işlemi yapılırken seçilen bölge içerisindeki en büyük değer alınır. Örneğin 1.bölgede bulunan 8, 7, 12, 9 değerleri içerisindeki en büyük değer 12 olduğu için alınmaktadır. Ortalama Havuzlama işlemi için ise seçilen bölgedeki değerlerin aritmetik ortalaması bulunarak havuzlama işlemi yapılır. Örneğin 2. bölgede 5, 3, 5, 7 rakamları için bölgenin aritmetik ortalaması 5 olduğundan matris değeri 5 olarak hesaplanır.

3. *Flattening Layer (Düzleştirme Katmanı)*: Düzleştirme katmanı evrimsel ve havuzlama katmanından aldığı verileri bu katmanda tek boyutlu diziye dönüştürür. Bu katmanda verilerin sınıflandırmaya hazır hâle getirilmesi sağlanır. Şekil 2.14'te düzleştirme katmanına yer verilmiştir.



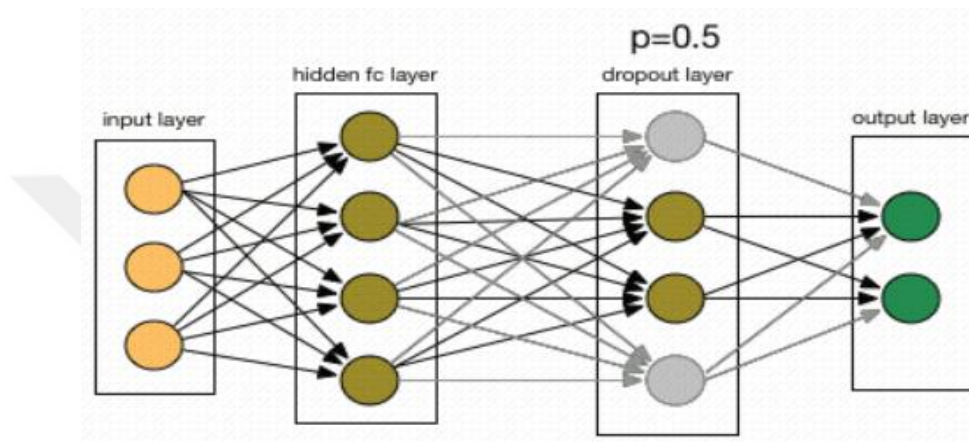
Şekil 2.14. Flatten katmanda düzleştirme işlemi (Akdağlı, 2021)

4. *Tam bağlantı (Fully Connected) katmanı*: Veriler düzleştirme katmanından alınarak sinir ağları ile öğrenme gerçekleştirilir. Katmanlardaki düğümlerde özellikler tutulur, ağırlık ve bias değiştirilerek öğrenme sağlanır. Bu katman CNN'deki son katman olarak bilinir.

Evrişim ve havuzlama katmanında ReLU aktivasyon fonksiyonu kullanılırken, tam bağlantı katmanında ise softmax fonksiyonu kullanılır. Olasılık değerleri [0-1] arasında değişir.

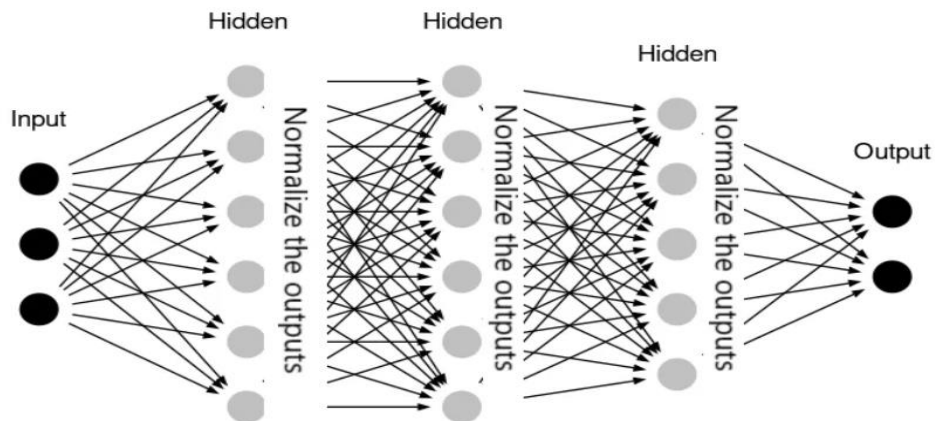
Bu katmanların haricinde kullanılan problemin türüne ve içeriğine göre *dropout* (*seyreltme*) ve *batch normalization* (normalizasyon) katmanı kullanılabilir.

5. *Seyreltme Katmanı (DropOut)*: Eğitimi sırasında nöronları rastgele kullanarak devre dışı bırakır. Bu da aşırı öğrenmeyi engeller. Aşırı öğrenmenin kullanılması ağın daha iyi öğrenmesini ve daha iyi performans göstermesini sağlar. Seyreltme katmanı *tam bağlantı* katmanından sonra kullanılır. Şekil 2.15'te Seyreltme katmanının gösterimine yer verilmiştir.



Şekil 2.15. Dropout katmanı gösterimi (Karaoğlu, 2021)

6. *Batch Normalization Katmanı*: Bu katman evrişim ve aktivasyon fonksiyonları arasında kullanılır. Bu katman sayesinde önceki katmanın öğrenmesi beklenmeden diğer katmanın öğrenmesi de başlayabilir. Yani senkron öğrenme imkânı sağlar. Gradyanların yok olmasına karşı durarak modelde iyi performans gerçekleşmesini sağlar. Seyreltme katmanı ile birlikte aşırı öğrenmeye karşı çıkar. Şekil 2.16'da toplu normalleştirme örneği gösterilmektedir.



Şekil 2.16. Toplu normalleştirme örneği (Rajagopal, 2018).

2.5.2. Derin Öğrenmede Transfer Öğrenme (Pre-Trained Modeller)

Makine öğrenmesi çoğu uygulamalarda başarılı bir şekilde uygulanırken bazı dezavantajları bulunmaktadır. Bunlardan en önde geleni eğitim aşamasının uzun zaman almasıdır. Bir modeli en baştan oluşturup eğitmek ve modele ince ayar yapmak oldukça zahmetli bir süreçtir. Bu problemi aşmanın en mantıklı yolu transfer öğrenmedir. Bu öğrenmede daha önceden tanımlanmış ve muhtemel ağırlıklar kullanılmaktadır. Başka bir deyişle transfer öğrenme bir görevde öğrenilen bilginin farklı bir görevde de kullanılması olarak tanımlanabilir.

Bu yöntem sayesinde makine öğrenmesine göre daha az veri ile daha yüksek başarımlar sağlanabilir. Transfer öğrenme az etiketi bulunan veya hiç etikete sahip olmayan veri kümelerinin modeldeki performansını artırmak için kaynak etki alanından etiketlenmiş eğitim verilerinde mevcut olan bilgiden yararlanmasını nedeniyle dünyada birçok uygulamada etkin olarak kullanılmaktadır (Day ve Khoshgoftaar, 2017).

Transfer adımı temel olarak 4 aşamada gerçekleşmektedir. İlk adımda model seçilir, genelde büyük veri kümeleri ile eğitilmiş modellerin seçimi avantaj oluşturmaktadır. 2. adımda transfer edilecek katmanlar belirlenir. Bu katmanın görevi verinin genel özelliklerini tanıyan eğitilmiş katmanlardır. 3. adım seçilen modelin son katmanı veri kümesine göre yeniden eğitilmelidir. Böylece yeni görevde gerekli olan özel nitelikleri öğrenir. Son adımda ise modelin performansı değerlendirilir. Model başarımı için ince ayar yapılabilir veya eğitim süresi artırılabilir (Fıkırkoca, 2024).

1) *Model seçimi:* Model seçmede amaca uygun bir mimari seçilmesi önemli olup MobilNet, Resnet, Vgg16/19 versiyonları buna örnektir. Bu modeller Imagenet gibi büyük veri kümeleri ile çalışıp oradaki ağırlıkları kullandığı için daha küçük veri setlerinde kullanıldığında en küçük spesifik özellikleri bile tespit etmektedir.

2) *Katmanların doldurulması:* Transfer öğrenme modellerin çoğunlukla ilk özellik çıkarma aşamasında katmanlar doldurulur. Doldurulan katmanlarda ağırlık güncellemesi olmaz. İlk katmanlarda genellikle kenar veya köşe gibi temel özellikler için kullanılır.

3) *Katman Ekleme:* Eklenen yeni katman modeli tahmin etmek ve öğrenmek için kullanılır. Yeni katmanlar daha önceden öğrenilmiş özellikleri baz alarak, özelleşmiş yapıları öğrenmeyi amaçlar. Genellikle çıktı ve sınıflandırma katmanı için katmanlar eklenir.

4) *Eklenen katmanların eğitme:* Eski ve yeni eklenen katmanlar yer değiştirildikten sonra model tekrardan eğitilir. Önceden eğitilen modellerin özellik çıkarma katmanı kullanılır. Eski özelliklerin korunmasının yanında yeni görev için sınıflandırma gücü geliştirilir.

5) *İnce ayar yapılması*: Modelin performansının daha iyi olması için kullanılır. İnce ayar model ağırlığının güncellenmesi, düşük öğrenme oranının kullanımı ve düzenleştirme gibi işlem aşamalarından oluşmaktadır. Kısacası hesaplama gücünün kısıtlı olması durumunda zaman açısından tasarruf etmek için eğitilmiş modeller son derece önemli olmaktadır.



3. MATERYALLER

3.1. Python

Python programlama dili dünyanın en hızlı büyüyen ve gelişme sağlayan programlama dilidir. Sadece yazılım ve bilgisayar mühendisleri değil, matematikçiler, ağ mühendisleri, veri bilimciler, istatistik ile uğraşan kişiler tarafından da kullanılmaktadır. İlk olarak 1991 yılında Guido Van Rossum tarafından piyasaya sürülen Python, günümüzde hâlâ geliştirilmekte olan yüksek seviyeli bir programlama dilidir. Bugün dünyadaki uygulamaların en az %27'sinin Python ile yazıldığı bilinmektedir. Açık erişimli bir dil olması geliştiriciler açısından önemlidir. Diğer dillerden ayrılan en önemli özelliği kolay öğrenimi, syntax yapısı ve diğer işletim sistemleri ile uyum içinde çalışabilmesidir. Bugün NASA, Google, Netflix, Spotify gibi birçok büyük şirket Python dilleri kullanmaktadır (“Python Programlama Dili Nedir? Nereelerde Kullanılır? – BlueMark Academy”, 2024).

Python'un kullanım alanı çok geniştir. Veri bilimi, yapay zekâ uygulamaları, makine ve derin öğrenme, veri analizi, mobil uygulamalar ve nesne tespiti gibi birçok alanda kullanım sağlanabilmektedir. Otomasyon işlemleriyle ilgilenen kullanıcılar da Python geliştirici olmasalar bile sürekli tekrar eden ve sıkıcı işler yapmak zorundaysa örneğin dosyaları ve klasörleri kopyalayıp bir sunucuya yüklemek işlemlerde basit bir Python kodu kullanarak zaman ve verim açısından tasarruf sağlayabilmektedirler. Python'da yazılım geliştirmek için komut satırına ve metin düzenleyiciye ihtiyaç duyulmaktadır. Ancak geliştiriciler için hız açısından da bir IDE'ye ihtiyaç bulunmaktadır. Bugün Eclipse, Vs Code, Pycharm gibi birçok IDE kullanıcılara seçenekler sunmaktadır. Python geliştiriciler açısından birçok alanda avantajlar sunmaktadır. Bunlardan en önemlisi *Object Oriented Programming* desteği ile içerisinde sunduğu güçlü kütüphane ve platforma sahip olmasıdır (Mustafa, 2024).

Bu kütüphaneler sayesinde aynı kodu tekrar tekrar yazmamıza gerek yoktur. Python'da en çok kullanılan kütüphaneler başlıca Pandas, Numpy, Tensorflow, SymPy, Keras ve Pytorch olarak bilinmektedir.

3.2. VS Stüdyo

Visual Stüdyo, Microsoft tarafından üretilen C, C++, Python, HTML gibi birçok dilde arayüz geliştirme, web geliştirme, konsol uygulamaları ve yapay zekâ uygulamaları gibi seçenek sunan bir IDE'dir. VS Stüdyo kod yazma, hata ayıklama, derleme gibi hizmetleri

kullanıcılara sunmaktadır. VS Studio özünde herhangi bir programlama dilini desteklemese de bunun yerine VSPackage olarak işlev sağlar. IDE bu açıdan 3 şekilde hizmet vermektedir. Bunlar sırasıyla SVsSolution, SVsUIShell ve SVsShell hizmetidir (“Microsoft Visual Studio”, 2024).

SVsSolution, Visual Stüdyo’nun çözüm kısmı olarak bilinmektedir. Bir çözüm birden fazla proje dosyası içerebilir. Çözüm gezgini penceresinden projeleri düzenlemek, eklemek bu aşamada SVsSolution’un görevidir. Bu aşama projelerin bir araya getirilip yönetildiği kısımdır.

SVsShell: Bu kısmında kod düzenleme, hata ayıklama, veri tabanı tasarımı gibi özellikler yer almaktadır. Ayrıca projelere ve çözümlere numaralandırma işlemleri sunulmaktadır. VSPackages kayıt ile ilgilenir, bu da eklentileri yönetmeye yarar.

SVsUIShell: Bu kısım VS Stüdyo’da pencereleme ve UI işlevselliği sağlar. Kullanıcıların arayüz işlemlerini sağlar.

3.3. Google Colab

Bir makine öğrenme aşamasında Google Colab, projenin başarıyla tamamlanmasında kritik bir rol oynar. Bu ortam veri bilimciler, mühendisler, yazılım geliştiriciler ve diğer ilgili ekipler arasında iş birliğini teşvik eder ve bilgi paylaşımını kolaylaştırır. Öncelikli olarak Google Colab, ekiplerin projenin her aşamasında bir araya gelerek fikir alışverişi yapmasını ve problemleri birlikte çözmesini sağlar. Veri bilimciler, geniş veri setlerini işleyerek modeller oluştururken mühendisler bu modelleri uygulamaya dökebilir ve geliştirebilir. Bu iş birliği sayesinde modeller daha hızlı iterasyonlarla geliştirilebilir ve daha yüksek doğruluk oranlarına ulaşılabilir.

İkinci olarak Google Colab, veri setlerinin ve kod tabanlarının paylaşılmasını kolaylaştırmaktadır. Bu, ekiplerin aynı veriye erişim sağlayarak tutarlı sonuçlar elde etmesini sağlar. Aynı zamanda, daha büyük ve karmaşık projelerde birden fazla ekip üyesinin aynı anda çalışmasına olanak tanır. Böylece projenin ilerlemesi hızlanır ve daha kapsamlı sonuçlar elde edilir.

Üçüncü olarak Google Colab, model performansını değerlendirmek için gerekli araçları sağlar. Bu araçlar genellikle modelin doğruluğunu, hatalarını, başarımlı metriklerini ve diğer önemli ölçümleri görselleştirmeye olanak tanır. Bu sayede ekipler, modelin gücünü anlayabilir, zayıf noktalarını belirleyebilir ve iyileştirmeler yapabilir.

Son olarak işbirlikçi bir ortam, ekip üyeleri arasında iletişimi güçlendirir. Anlık mesajlaşma araçları, toplantılar ve paylaşılan belge depoları gibi araçlar, ekiplerin birbirleriyle

düzenli olarak iletişim kurmasını ve proje hedeflerini net bir şekilde belirlemesini sağlar. Bu da ekip içi uyum ve motivasyonu artırır, dolayısıyla projenin başarı şansını artırır.

Bugün dünya üzerinde yapılan yazılımlarda en çok kullanılan programlama dili Python olmakla birlikte, Ruby, Octave gibi diller de kullanılmaktadır. Bu diller aracılığıyla yapılan yazılımlar çeşitli derleyicilerde çalıştırıldığı gibi Google Colab platformunun kullanıcılara ücretli veya ücretsiz GPU desteği ve sistem ram seçenekleri sunması derin öğrenme modellerinde daha hızlı sonuç alınmasını ve kısa sürede eğitim işlemlerinin bitirilmesini sağlamaktadır (Çetin, 2020).

3.4. Kütüphaneler

3.4.1. Pandas

Pandas, veri manipülasyonu ve analizi için yaygın olarak kullanılan bir Python kütüphanesidir. Pandas'ta iki temel veri yapısı bulunmaktadır. Eğer veriler tek boyutlu ise seriler, çok boyutlu ise Veri çerçevesi kullanılır. Bu iki yapı mevcut verilerin tutulması ve analiz edilip işlenmesi için seçenek sağlar. Projelerde veri setlerinin okunması, işlenmesi ve analiz edilmesi için Pandas kullanılmaktadır. Pandas, Numpy'ın üzerine inşa edilmiş bir pakettir. Diğer 3. parti kütüphanesiyle bilimsel işlem ortamına iyi bir şekilde entegre olmayı amaçlar (Şahin, 2020). Veri çerçevesi yapısıyla verilerin düzenlenmesi ve işlenmesi kolaylaşır, bu da veri analizi ve model eğitimi süreçlerini hızlandırır. Pandas, kullanıcıları için önemli bir kütüphane olup veri yapılarında, veri manipülasyonlarında, verinin temizlenmesinde, veri analizi ve görselleştirilmesinde kullanılmaktadır. Ayrıca çeşitli sorunlara çözümler bulunmasını sağlamakta olup kapsamlı ve kullanıcı dostu bir dokümantasyona sahiptir.

3.4.2. NumPy

NumPy, bilimsel hesaplama ve çok boyutlu dizilerin işlenmesi için kullanılan temel bir Python kütüphanesidir. Projede, verilerin matematiksel işlemlerinin ve dizi manipülasyonlarının gerçekleştirilmesi için NumPy kullanılabilir. Özellikle matris işlemleri ve vektör hesaplamaları gibi derin öğrenme modelleri için temel işlemlerde kullanılır. Numpy dizileri hız ve işlev açısından farklıdır. Python dizilerinden farklı olarak Numpy'de diziler homojen olmalıdır. Yani tüm veri tipi aynı olmalıdır (Durna, 2019).

Numpy çok boyutlu diziler için optimum seçenek olup matematiksel işlemler için çoklu destek sağlamaktadır. Matris işlemleri, transpozunu alma, matris terimi bulma gibi işlemler numpy sayesinde gerçekleşmektedir. Bu kütüphane Python diline entegre olup uygulamalar “pip install numpy” komutu ile kolayca yüklenebilmektedir (Asutay, 2023).

3.4.3. Matplotlib

Matplotlib, veri görselleştirme için kullanılan bir Python kütüphanesidir. Projede, verilerin grafiklerle görselleştirilmesi ve analiz edilmesi için Matplotlib kullanılır. Grafikler ve görseller, veri setlerinin anlaşılmasını ve model performansının görsel olarak takip edilmesini kolaylaştırır. Veri görselleştirme eldeki soyut verilerin somutlaştırılması ve bundan bir ön bilgi edilme noktasında önemlidir. Veri görselleştirme işleminde grafiğin anlaşılır olması ve renklerin uyumlu olması önemlidir. Matplotlib sayesinde 2 veya 3 boyutlu çizimlerin yapılması mümkündür (Alabaş, 2019). Ayrıca Matplotlib sayesinde çıktıları Tkinter, Qt gibi eklentilerle uyumlu çalışarak uygulamalara nesne tabanlı API sağlamaktadır.

3.4.4. Seaborn

Seaborn, Matplotlib'e dayalı bir veri görselleştirme kütüphanesidir ve daha çekici ve bilgilendirici grafikler oluşturmak için kullanılır. Normalde veri görselleştirme işlemi hem Matplotlib hem de Seaborn ile yapılmaktadır. Fakat Seaborn kütüphanesinde daha az kod ile daha çok işlem yapıldığından kullanıcılara cazip gelmektedir (Balık, 2018). Projede, veri setlerinin daha karmaşık grafiklerle görselleştirilmesi ve analiz edilmesi için Seaborn kullanılabilir. Özellikle karmaşık ilişkileri gösteren heatmap'ler gibi görseller oluşturmak için kullanışlıdır.

3.4.5. Scikit-learn

Scikit-learn, Python'da makine öğrenimi modellerinin oluşturulması, eğitilmesi ve değerlendirilmesi için kullanılan bir kütüphanedir. Projede model seçimi, eğitimi, hiper parametre ayarı ve sonuçların değerlendirilmesi için Scikit-learn kütüphanesi kullanılmaktadır. Scikit-learn 3 API'den oluşur. Estimator API, bu yapı datayı öğrenebilen sınıflara denir. Transformer API veri setinde değişiklik ve filtreleme yapabilen tahmin edicilerdir. Son olarak Predictor API ise modelin tahminlenmesi için kullanılır. Predict metodu daha önce öğrenilmemiş data üzerinde tahminleme de yapar (Koçulu, 2021). Sınıflandırma, regresyon, kümeleme ve boyut azaltma gibi birçok makine öğrenimi algoritmasını içermektedir.

3.4.6. Keras

Keras, yüksek seviyeli bir derin öğrenme API'si olarak bilinmektedir ve TensorFlow, Theano veya Microsoft Cognitive Toolkit (CNTK) gibi altta yatan sayısal hesaplama kütüphaneleri üzerine kuruludur. Günümüzde 200.000 kullanıcıya sahip ve start-uplardan oluşan büyük şirketler bu kütüphaneyi tercih etmektedir. Keras, derin öğrenme modellerinin hızlı bir şekilde indirilmesi, oluşturulması, eğitilmesi, girdilerin sinir ağına hazırlanması ve

değerlendirilmesi için kullanılır (Cobanov, 2019). Yine CPU ve GPU’da modelleri sorunsuz çalıştırma olanağı vermektedir. CNN (Evrişimli Sinir Ağı) ve RNN (Yinelemeli Sinir Ağı) türlerini desteklemektedir.

3.4.7. TensorFlow

TensorFlow, Google tarafından geliştirilen açık kaynaklı bir makine öğrenimi ve derin öğrenme kütüphanesidir. TensorFlow derin öğrenme için geliştirilmiş açık kaynaklı bir matematik kütüphanesidir (Türkoğlu, 2021). Projede, TensorFlow'un Keras ile birlikte kullanılması, derin öğrenme modellerinin oluşturulması, eğitilmesi ve dağıtılmasını kolaylaştırır. Bu kütüphane başlangıç aşamasında bir API sunmaktadır. Böylece uygulamanın performansında yükselim sağlanır. TensorFlow'un yüksek performansı, dağıtık hesaplama yetenekleri ve geniş topluluk desteği, projenin derin öğrenme kısmında tercih edilmesinin nedenleri arasında yer aldığı söylenebilir.

3.4.8. Kivy

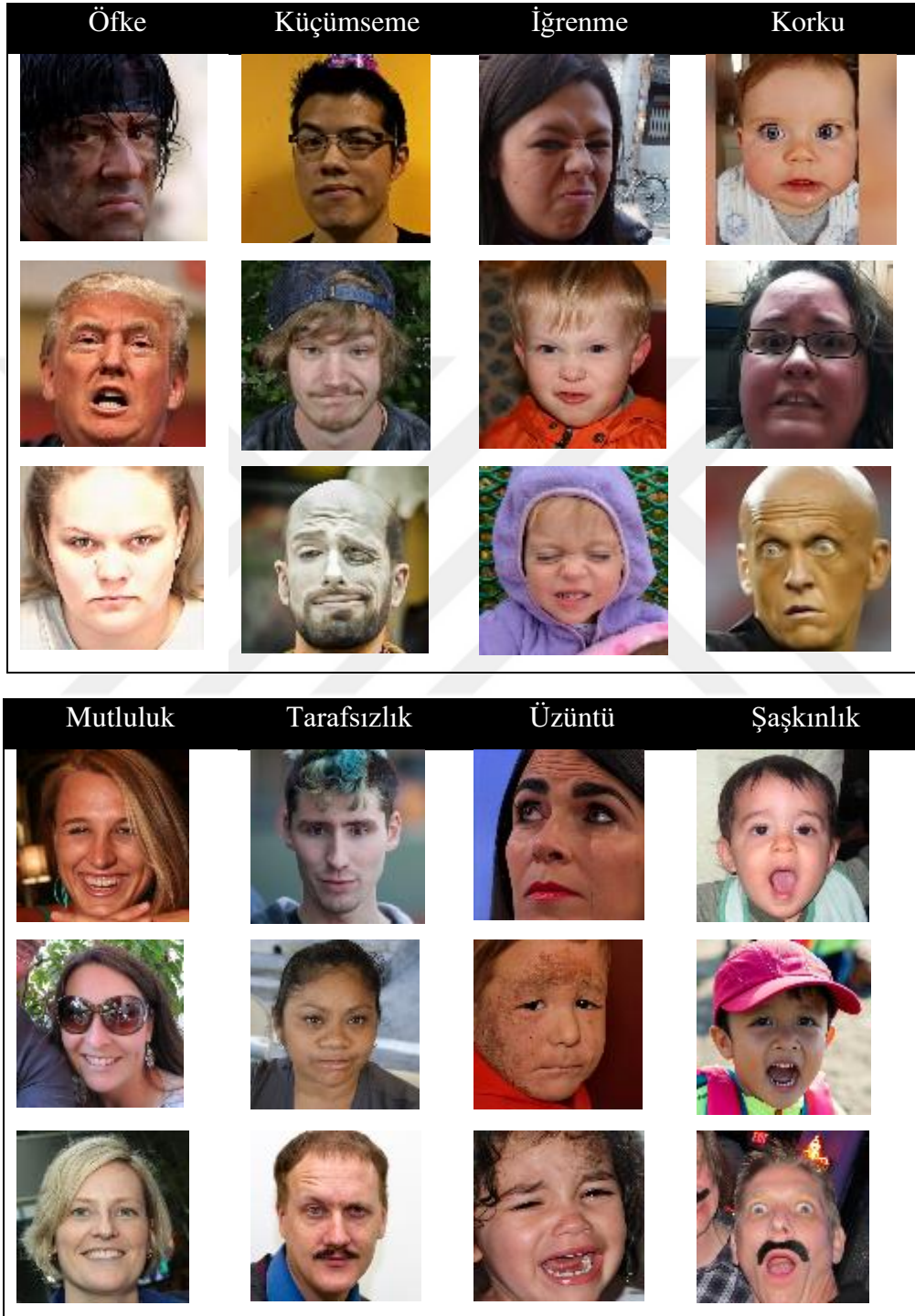
Kivy, Python dilinde açık kaynaklı olarak geliştirilen bir kullanıcı arayüzü (UI) kütüphanesidir. Android’den Windows’a kadar birden fazla işletim sistemini destekleyen bir grafik arayüze sahip olup çapraz platformlar için uygulama geliştirmede yardımcı olmaktadır (Uğur, 2020). Kodların %90’ı Python ile oluşturulmaktadır. Bu kütüphane, etkileyici ve modern kullanıcı arayüzleri oluşturmayı sağlar ve çoklu platform desteği sunar. Kivy' nin temel özellikleri arasında dokunmatik arayüz desteği, animasyonlar, olay yönetimi ve UI bileşenlerinin esnek kullanımı bulunur. Kivy, Python dilini kullanarak hızlı ve kolay bir şekilde platformlar arası uyumlu mobil uygulamalar ve masaüstü uygulamalar geliştirmeyi sağlar. Ayrıca, Kivy' nin açık kaynaklı olması ve geniş bir topluluğa sahip olması, geliştiricilerin desteğe erişimini ve sürekli olarak gelişen bir ekosistemde yer almasını sağlamaktadır.

3.5. Veri

Kullanılan veri seti Kaggle platformundan alınmıştır. Facial Expression Training Data adlı veri setinde sinir ağları için işlenmiş AffectNet, yüz ifadeleri ile etiketlenmiş büyük bir veri tabanıdır. Yaygın bellek kısıtlamalarına uyum sağlamak için çözünürlük 96x96'ya düşürülmüştür. Bu, tüm görüntülerin tam olarak 96x96 piksel olduğu anlamına gelir.

Veri setinde tek renkli görüntüler bulunmamaktadır. Her resmin tekil değer ayrıştırma metodu ile temel bileşen analizi hesaplanmıştır. Temel bileşenler içindeki ilk bileşenin yüzdesi (0. indeks) için eşik değeri %90'dan düşük olarak ayarlanmıştır. Bu, çoğu, hatta tüm tek renkli görüntülerin filtrelenmiş olduğu anlamına gelmektedir. Veri setimiz, sekiz temel duygu

kategorisini içermektedir. Bu sınıfların veri setinden örnek resimlerini ve veri setindeki bulundukları miktarları aşağıdaki tablolarda görmek mümkündür (Segal, 2023). Şekil 3.1’de veri setindeki görüntü ve sınıflarına yer verilmiştir.



Şekil 3.1. Veri seti örnek görseller

Çizelge 3.1’de veri setindeki sınıfların ve içerdiği görüntülerin dağılımı verilmiştir.

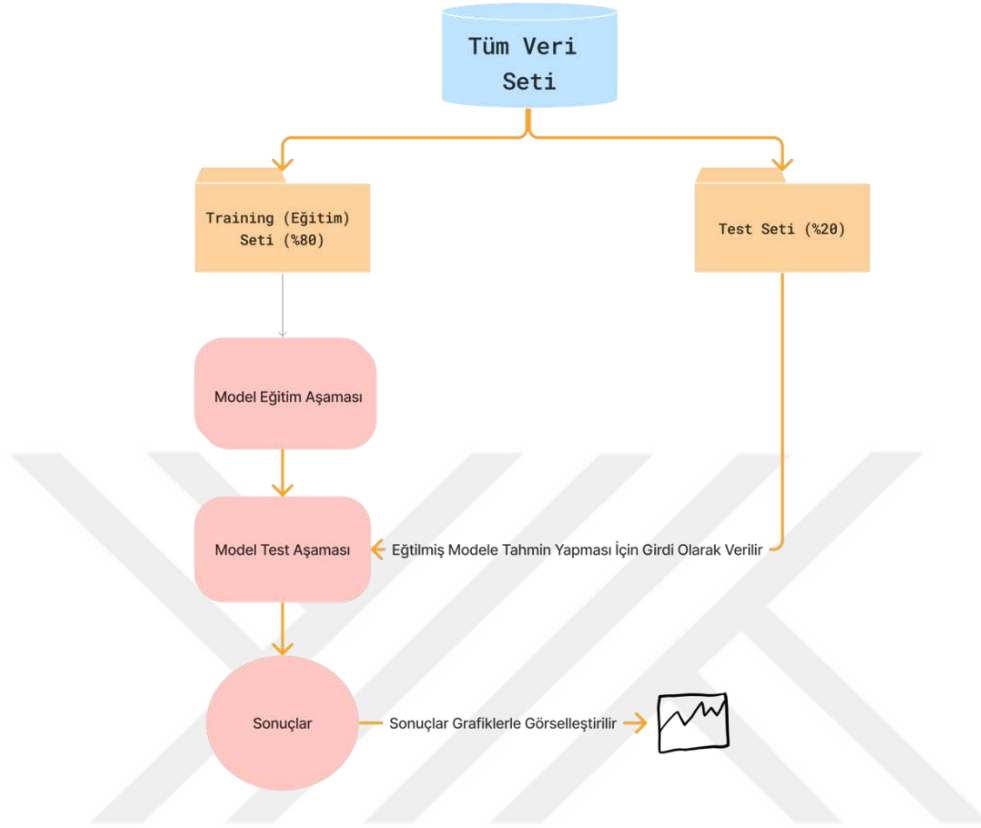
Çizelge 3.1. Veri seti sınıf ve miktar tablosu

Sınıf	Veri Miktarı
Şaşkın	4039
Korku	3176
Tarafsız	5126
Üzgün	3091
İğrenç	2477
Küçümseme	2871
Mutlu	5044
Öfke	3218

Elimizdeki resimler modellere hazır hâle getirilmeden önce 96*96’dan 224*224’lük resimlere dönüştürülmüştür. Çalıştırılan her model için resim girdileri ön işlem aşamasından geçirilmiştir. Bu ön işlem aşaması, modele verilecek olan görüntünün nümerik olarak doğru formata sahip olmasını ve modelin eğitiminin, tahmin süreçlerinin doğru şekilde gerçekleşmesini sağlamaktadır. Bu aynı zamanda, model eğitimi sürecinde görüntülerin uygun şekilde işlenerek kullanılmasını ifade etmektedir. Eğitimde kullanılan test veri setleri tüm veri setinin 20%’si olacak şekilde ayarlanmıştır. Veri setimizde toplamda 29.042 görüntü yer almaktadır, fakat bunların 20.518’i kullanılarak modellerin eğitim ve test işlemleri gerçekleştirilmiştir. 8 duyguyu tahmin etmedeki doğruluk değerleri proje sonuçlarını tatmin etmediğinden dolayı daha iyi sonuçlar alabileceğimiz 5 sınıfın kullanılması tercih edilmiştir. Bu 8 sınıftan küçümseme ve iğrenme sınıfları veri azlığından dolayı veri setinden çıkartılmıştır. Korku ve üzüntü sınıfları benzer veriler içermesinden dolayı bir sonraki aşamada korku sınıfı da veri setimizden çıkartılmıştır. Veri setinde en az veriye sahip sınıflardan 2’si çıkarılırken çok veriye sahip sınıflar eğitim için kullanılmıştır.

Modeller eğitilip test edilmeden önce, verilerin eğitim ve test verisi olarak ikiye ayrılması gerekiyordu. Bu adımın temel amacı, modeli eğitirken kullanılan verilerle aynı veri kümesi üzerinde doğrulama yapmak yerine, modelin gerçek dünya verileri üzerinde ne kadar iyi performans göstereceğini değerlendirebilmektir. Eğitim verisi, modelin öğrenme sürecinde kullanılan veri kümesini temsil ederken test verisi ise modelin eğitim sonrası performansını değerlendirmek için ayrılan veri kümesidir. Bu sayede modelin genelleştirme yeteneği ve gerçek dünya verilerine uyumluluğu daha sağlıklı bir şekilde değerlendirilmektedir.

Aşağıdaki şekilde bu aşamanın nasıl gerçekleştiği gösterilmektedir. Şekil 3.2 verinin ayrılma yüzdesini ve eğitim aşamasının şeklini inceleyebilirsiniz.



Şekil 3.2. Veri seti modele hazırlama aşaması

3.6. Yöntemler

3.6.1. DenseNet121

DenseNet121, yoğun evrişimli ağ anlamına gelmektedir ve derin öğrenme alanında kullanılan bir yapay sinir ağı modelidir. Bu model, ResNet gibi diğer modellerden farklı olarak her katmanın tüm önceki katmanlardan ek girdiler almasını ve kendi özellik haritalarını sonraki tüm katmanlara iletilmesini sağlamaktadır. DenseNet, görsel nesne tanıma problemleri için kullanılan yeni bir derin öğrenme teknolojisidir ve farklı yapılar sahip olan DenseNet-121, DenseNet-160 ve DenseNet-201 gibi çeşitli versiyonları bulunmaktadır. Bu çalışmada kullanılan versiyon DenseNet-121'dir. Bu model, karmaşık görsel veri setlerinde yüksek performans sağlamak amacıyla tasarlanmıştır (Huang, Liu, van der Maaten ve Weinberger, 2018). DenseNet121 modelinin oluşturulmasında dikkate alınan parametreler aşağıdaki gibidir:

1. *Giriş şekli (input shape)*: Görsellerin boyutu 224*224 ve 3 kanallı olarak belirlenmiştir. Yani resmin her pikselinde 3 farklı renk RGB(kırmızı, yeşil, mavi) yer almaktadır.

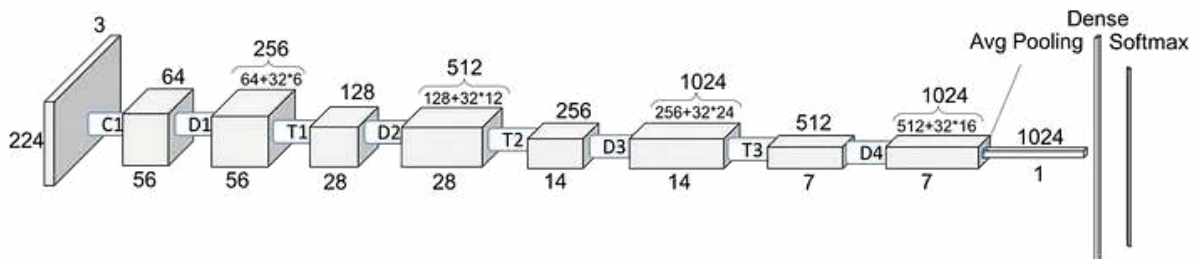
2. *Ağırlıklar (weights)*: Modelin ağırlıkları, ImageNet veri kümesinde önceden eğitilmiş ağırlıklar olarak belirlenmiştir. Böylece, transfer öğrenme yöntemi kullanılarak modelin önceden eğitilmiş ağırlıkları verilmiştir.

3. *include_top*: Bu parametre, modelin üst katmanlarını (tam bağlantı katmanları) dâhil etmeme seçeneği sunarak resimlerden özellik çıkarmak için modelin kullanılmasını sağlamıştır.

4. *pooling='max'*: Modeldeki havuzlama (pooling) katmanının maksimum havuzlama işlemi yapacağını belirtmektedir. Bu, özellik haritalarının en büyük değerlerini alarak özellik vektörlerini oluşturmaktadır. Dolayısıyla resimlerden özellik çıkarılırken daha hızlı ve efektif bir özellik havuzu oluşturulmasını sağlamıştır.

5. *base_model.trainable*: Bu parametre *true* olarak ayarlandığından, modelin eğitim sırasında ağırlıklarının güncellenmesine izin verilerek modelin eğitildikçe ağırlıklarının veriye uyumunu sağlamıştır.

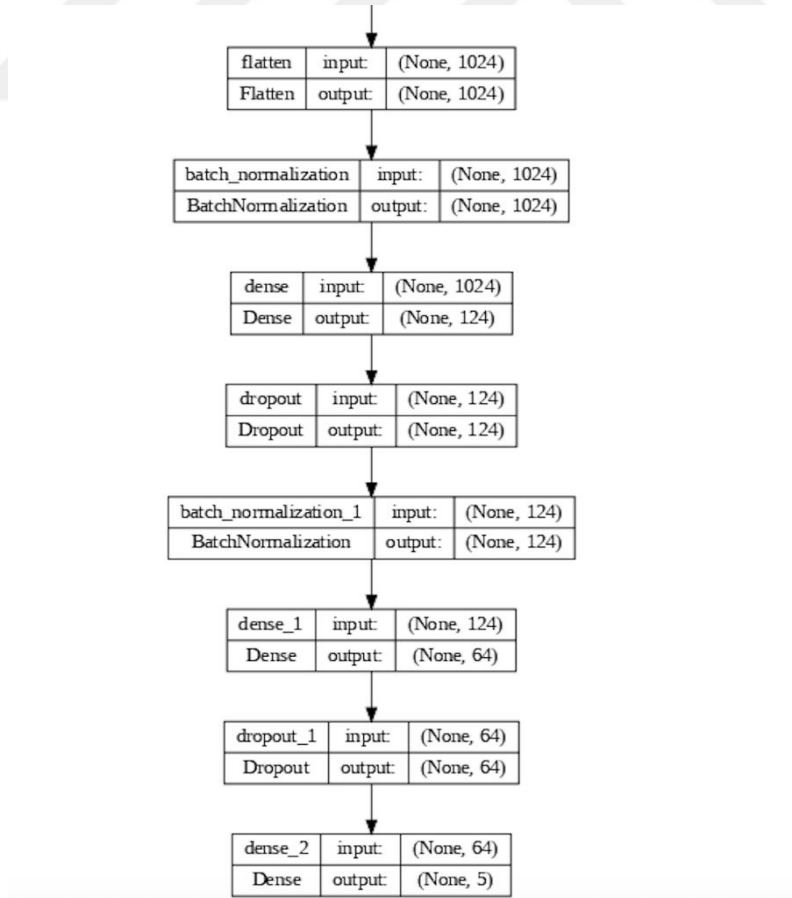
DenseNet'in temel yapısı, yoğun bir şekilde bağlı katmanlardan oluşur. Bu yapı, her katmanın önceki tüm katmanlardan ek girdiler almasını ve kendi özellik haritalarını sonraki tüm katmanlara iletmesini sağlar. Ayrıca, DenseNet'in avantajları arasında güçlü gradyan akışı, parametre ve hesaplama verimliliği, daha çeşitli özellikler ve düşük karmaşıklık seviyelerini koruma gibi özellikler bulunmaktadır. DenseNet ayrıca, ResNet gibi diğer modellere göre daha küçük boyutlu olmasıyla da dikkat çekmektedir. Bu modelde her katman önceki tüm katmanlardan girdi almakta ve bu da daha zengin desenlerin elde edilmesine olanak tanımaktadır. Son katmanda global ortalama havuzlama (average pooling) işlemi gerçekleştirilmekte ve ardından bir softmax sınıflandırıcı eklenmektedir. Şekil 3.3'te DenseNet-121 modelinin yapısı gösterilmiştir.



Şekil 3.3. DenseNet-121'in şeması (Huang ve diğerleri, 2018).

Uygulamada kullanılan ve temel model olarak alınan modelin üzerine eklenen katmanlar aşağıdaki gibidir:

1. Temel modelin çıktısı alınır.
2. Çıktıya düzleştirme işlemi ile düzleştirilir.
3. Düzleştirilmiş çıktıya normalizasyon uygulanır.
4. Düzleştirilmiş ve normalleştirilmiş çıktı, ReLU aktivasyon fonksiyonu ile Dense (124) tam bağlantılı bir katmana iletilir.
5. Ağdaki aşırı uyumu azaltmak için Dropout (0,7) uygulanır.
6. BatchNormalization() ile normalleştirme tekrar uygulanır.
7. Daha sonra, ReLU aktivasyon fonksiyonu ile bir Dense(64) tam bağlantılı katman eklenir.
8. Aşırı uyumu azaltmak için tekrar Dropout(0,4) uygulanır.
9. Son olarak, softmax aktivasyon fonksiyonu ile bir Dense(5) çıktı katmanı eklenir, burada 5 farklı duygu sınıfını tahmin etmek için kullanılır. Aşağıda DenseNet-121 modeline eklenen katmanlar gösterilmiştir.



Şekil 3.4. DenseNet-121'e eklenen katman şeması

Derin öğrenme modellerinin eğitimi sırasında belirli noktalarda çağrılan ve modelin davranışını değiştirmek veya kontrol etmek için kullanılan işlevler vardır. Model eğitilirken kullanılan yöntemler şunlardır:

1. ModelCheckpoint: Eğitim safhasında modeli her aşamada kontrol eden bir uyarıcıdır. Modelin en iyi performans gösterdiği turları kaydetmek için kullanılır. save_best_only=True parametresi, sadece en iyi performans gösteren modelin kaydedilmesini sağlar.
2. ReduceLROnPlateau: Bu işlev, eğer modelin performansı belirli bir süre boyunca gelişmiyorsa (patience), modelin öğrenme hızını azaltır. Monitor parametresi, izlenecek metriği belirtir, burada kayıp değeri izlenir. Factor=0,1 parametresi, öğrenme hızının azaltılacağı oranı belirtir. Patience=2 parametresi, öğrenme hızının azaltılmasından önce beklenilecek tur sayısını belirtir.
3. EarlyStopping: Bu işlev, eğer model belirli bir süre boyunca (patience) belirli bir metrik gelişme göstermiyorsa, eğitimi durdurmaktadır. Monitor parametresi, izlenecek metriği belirtir. Çalışmada validation (doğrulama) veri setindeki doğruluk metriği izlenerek modelin eğitiminin durdurulması sağlanmıştır. Patience'i 5 belirleyerek, eğitimin 5 tur boyunca doğruluk değeri artmadığı zaman durdurulması sağlanmıştır.

Model oluşturulduktan ve bu işlevler belirlendikten sonra model derleme aşamasına geçer. Derleme aşamasında belirlenmesi gereken birkaç parametre vardır. Çizelge 3.2'de bu parametreler gösterilmektedir.

Çizelge 3.2. DenseNet-121 derleme aşamasındaki kullanılan parametreler

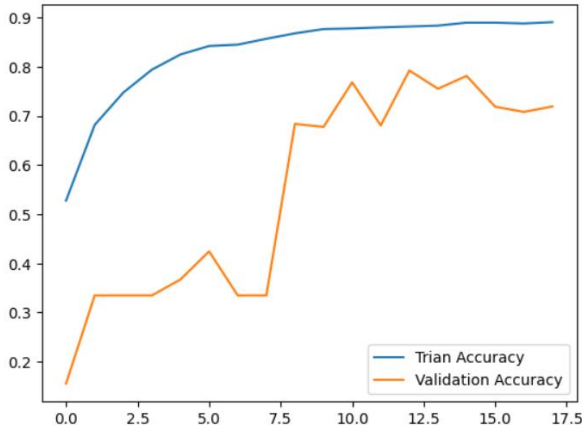
Model	DenseNet-121
Optimize edici	Adam
Öğrenme oranı	0,001
Kayıp fonksiyon	Categorical Crossentropy
Metrik	Doğruluk

Modelin compile edilmesi yani derlenmesinden sonra modelin eğitim aşamasına geçilir. Bu aşamada kullanılan parametre ve veri miktarları Çizelge 3.3'te verilmiştir.

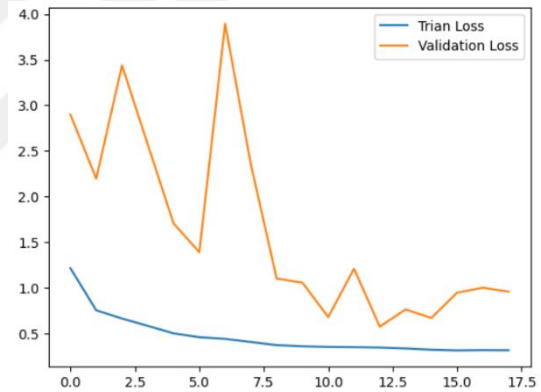
Çizelge 3.3. DenseNet-121 eğitim parametreleri ve geri çağırımlar

Eğitim Verisi (Train)	(16414, 224, 224, 3) (16414, 5)
Yığın boyutu	128
Eğitim sayısı	30
Test Verisi (Validation)	(4104, 224, 224, 3) (4104, 5)
Yöntemler	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

Transfer Learning yöntemiyle eğitilen Densenet121 modelinin eğitim grafikleri, loss (kayıp) metriği ile modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu göstermektedir. Bu grafiğin düşüşü, modelin eğitim sürecinde kayıpların nasıl azaldığını yansıtırken accuracy (doğruluk) metriği, modelin verilen veri kümesinde ne kadar doğru tahminler yaptığını göstermektedir. ROC (Receiver Operating Characteristic) eğrisi ise modelin sınıflandırma performansını değerlendirmek için kullanılır. Şekil 3.5 ve Şekil 3.6’da modele ait doğruluk ve kayıp değerlerini gösteren grafiğe yer verilmiştir.



Şekil 3.5. DenseNet-121 doğruluk değerleri



Şekil 3.6. DenseNet-121 kayıp değerleri

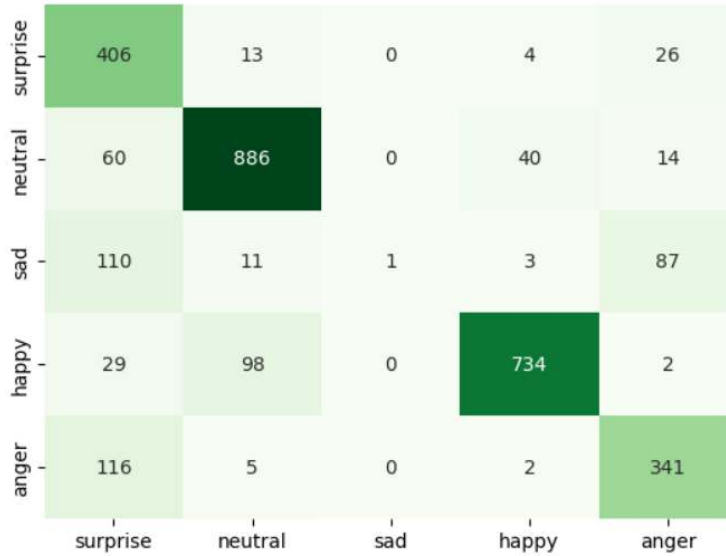
DenseNet-121 modelinde eğitim sonunda bazı sonuçlara ulaşılmıştır. Eğitim boyunca eğitim doğruluğu sürekli artarken bu oran %89'a kadar çıkmıştır. Doğrulama tarafında da ise bu oran 13.eğitim sonunda %79 olarak bulunmuştur. Modelde eğitim kaybı 18.turda 0,3178'e kadar inerken, doğrulama kaybında büyük bir düşüş gözlemlenmemiştir. En düşük 0,5788'e kadar düşen doğrulama kaybı 18. turda 0,9589'a kadar yükselince model burada erken durdurma yapmıştır. Modelin aşırı öğrenmeye en yakın olduğu durum 4.turda'dır. Çünkü eğitim doğruluğu hızla artarken doğrulama doğruluğu %33'ler civarında seyretmiştir. Modelde herhangi bir eksik öğrenme problemi ise gözükmemektedir. Eğitim ve doğrulama doğruluğu

arasında aşırı bir fark yok ve doğrulama doğruluğu %70'lerin üzerindedir. Precision (kesinlik), recall (duyarlılık) ve f-1-skor değerleri Çizelge 3.4'teki gibidir.

Çizelge 3.4. DenseNet-121 metrik tablosu

Doğruluk	0,7925	Kayıp	0,5788
	Kesinlik	Duyarlılık	F1-score
Sürpriz	0,56	0,90	0,69
Doğal	0,87	0,89	0,88
Üzgün	1,00	0,00	0,01
Mutlu	0,94	0,85	0,89
Sinirli	0,73	0,73	0,73
Makro ortalama	0,82	0,68	0,64
Ağırlık ortalama	0,83	0,79	0,77

Bu veriler haricinde her bir sınıftan ne kadar sayıda doğru sınıfa atama yapıldığını ve yanlış atamaların hangi sınıflara yapıldığı Şekil 3.7'de gösterilmektedir.



Şekil 3.7. DenseNet-121 karmaşıklık matrisi

ROC eğrisi (Receiver Operating Characteristic curve), sınıflandırma problemlerinde modelin performansını değerlendirmek için kullanılan önemli bir araçtır. ROC eğrisi, duyarlılık (sensitivity) ve özgüllük (specificity) arasındaki ilişkiyi gösteren bir grafikdir.

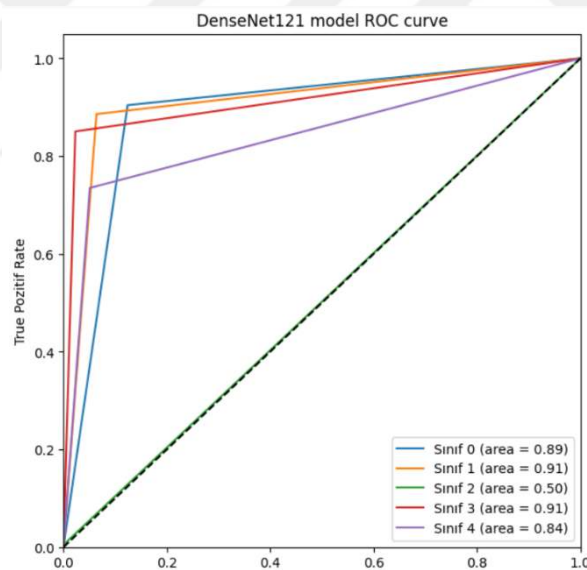
ROC eğrisi, bir sınıflandırma modelinin doğruluk performansını değerlendirirken yanlış pozitif oran (false positive rate) ve doğru pozitif oran (true positive rate) arasındaki ilişkiyi

görselleştirmektedir. Yanlış pozitif oran, esasında negatif olan ancak model tarafından pozitif olarak sınıflandırılan örneklerin oranını ifade ederken doğru pozitif oran ise gerçek pozitif olarak sınıflandırılan örneklerin oranını ifade etmektedir.

ROC eğrisi, x-ekseninde yanlış pozitif oranın, y-ekseninde ise doğru pozitif oranın yer aldığı bir grafik olarak çizilir. İdeal bir sınıflandırıcı, ROC eğrisinin sol üst köşesine daha yakın olacaktır.

ROC eğrisinin altında kalan alan (area under the curve-AUC), sınıflandırıcının performansını tek bir sayısal değerle ölçmek için kullanılır. AUC değeri, 0 ile 1 arasında bir değer alır ve 1'e ne kadar yakınsa sınıflandırıcının performansı o kadar iyidir. AUC değeri 0,5 ise, sınıflandırıcının rastgele sınıflandırma yaptığı anlamına gelir.

ROC eğrisi, özellikle dengesiz sınıflandırma problemlerinde (yani sınıflar arasında belirgin bir dengesizlik olduğunda) modelin performansını değerlendirmek için yaygın olarak kullanılır. Ayrıca, farklı kesme noktalarında duyarlılık ve özgüllük arasındaki trade-off'ları görselleştirmek için de kullanışlıdır. Densenet121 modeli için oluşan ROC eğrisi grafiği Şekil 3.8'de gösterilmektedir:



Şekil 3.8. DenseNet-121 ROC eğrisi

3.6.2. EfficientNet

EfficientNet, bir evrişimli sinir ağı mimarisi ve ölçekleme yöntemidir. Bu yöntem, derinlik/genişlik/çözünürlük boyutlarını bir bileşik katsayı kullanarak eşit şekilde ölçeklendirmektedir. Geleneksel uygulamalardan farklı olarak EfficientNet ölçekleme yöntemi ağırlık, derinlik ve çözünürlüğü sabit ölçekleme katsayıları kümesiyle eşit şekilde ölçeklemektedir.

EfficientNet' in temel özellikleri şunlardır:

1. Bileşik ölçekleme: Genişlik, derinlik ve çözünürlük boyutlarını eşit şekilde ölçeklendirerek mükemmel bir denge sağlar.
2. AutoML ve bileşik ölçekleme: AutoML Mobile çerçevesi ile geliştirilen mobil boyutlu temel ağ olan EfficientNet-B0, bileşik ölçekleme yöntemi ile geliştirilerek EfficientNet-B1'den B7'ye kadar iyileştirilir.
3. Yüksek doğrulukta, EfficientNet-B7, 66M parametre ve 37B işlem ile ImageNet'te %84,4 üst-1 / %97,1 üst-5 doğruluğa ulaşır.
4. Hız ve boyut farkı: Önceki lider GPipe'e kıyasla EfficientNet, CPU çıkarımında 8,4 kat daha küçük ve 6,1 kat daha hızlıdır.
5. Model çeşitliliği: Farklı ölçekleme katsayılarına sahip çeşitli modeller sunar (EfficientNet-B0, EfficientNet-B1 vb.). Böylece kullanıcılar ihtiyaçlarına en uygun model varyantını seçebilirler (Tan ve Le, 2019) ("What is EfficientNet?", 2023).

Bu uygulamada seçilen EfficientNet-B5'in eğitim aşamasındaki girdi verileri aşağıdaki gibidir:

1. Giriş şekli (input shape): Görsellerin boyutu ve RGB renk uzayında (224, 224, 3) olarak belirlenmiştir. Bu, 224x224 piksel boyutunda ve 3 kanallı (RGB) bir giriş şekli olduğundan verideki resimlere uygun olarak belirlenmiştir.

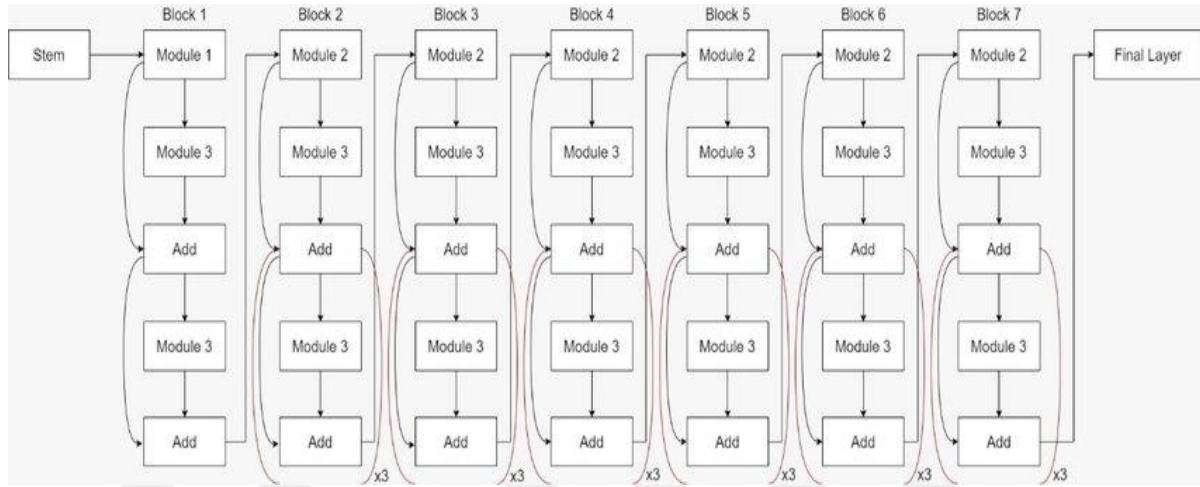
2. Ağırlıklar (weights): Modelin ağırlıkları, ImageNet veri kümesinde önceden eğitilmiş ağırlıklar olarak belirlenmiştir. Böylece, transfer öğrenme yöntemi kullanılarak modelin önceden eğitilmiş ağırlıkları aktarılmıştır.

3. include_top: Bu parametre, modelin üst katmanlarını (tam bağlantı katmanları) dâhil etmeme seçeneği sunarak resimlerden özellik çıkarmak için modelin kullanılmasını sağlamıştır.

4. base_model.trainable: Bu parametre *true* olarak ayarlandığından, modelin eğitim sırasında ağırlıklarının güncellenmesine izin verilerek modelin eğitildikçe ağırlıklarının veriye uyumunu sağlamıştır.

EfficientNetB5, yoğun bir şekilde bağlı katmanlardan oluşur. Bu yapı, her katmanın tüm önceki katmanlardan ek girdiler almasını ve kendi özellik haritalarını tüm sonraki katmanlara iletilmesini sağlar. Ayrıca, EfficientNet'in avantajları arasında güçlü gradyan akışı, parametre ve hesaplama verimliliği, daha çeşitli özellikler ve düşük karmaşıklık seviyelerini koruma gibi özellikler bulunmaktadır. EfficientNet ayrıca, diğer modellere göre daha küçük boyutlu olmasıyla da dikkat çekmektedir. Bu modelde, her katman önceki tüm katmanlardan girdi almaktadır. Bu da daha zengin desenlerin elde edilmesine olanak sağlamaktadır. Son katmanda

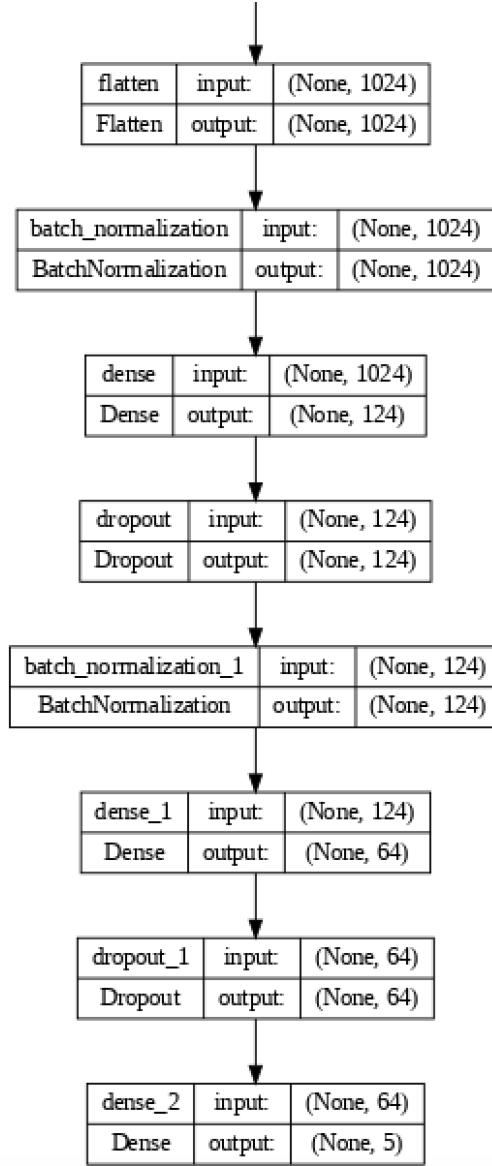
global ortalama havuzlama (average pooling) işlemi gerçekleştirilir ve ardından bir softmax sınıflandırıcı eklenir. Şekil 3.9’da EfficientNet-B5 şeması gösterimi yer almaktadır.



Şekil 3.9. EfficientNet-B5 şeması

Uygulamada kullanılan ve temel model olarak alınan modelin üzerine eklenen katmanlar aşağıdaki gibidir:

1. Temel modelin çıktısı alınır.
2. Çıktı düzleştirme işlemi ile düzleştirilir.
3. Düzleştirilmiş çıktıya normalleştirme uygulanır.
4. Düzleştirilmiş ve normalleştirilmiş çıktı, ReLU aktivasyon fonksiyonu ile Dense(124) tam bağlantılı bir katmana iletilir.
5. Ağdaki aşırı uyumu azaltmak için Dropout(0,7) uygulanır.
6. BatchNormalization() ile tekrar normalleştirme uygulanır.
7. Daha sonra, ReLU aktivasyon fonksiyonu ile bir Dense(64) tam bağlantılı katman eklenir.
8. Aşırı uyumu azaltmak için tekrar Dropout(0,4) uygulanır.
9. Son olarak, softmax aktivasyon fonksiyonu ile bir Dense(5) çıktı katmanı eklenir, burada 5 farklı duygu sınıfını tahmin etmek için kullanılır. Şekil 3.10’da EfficientNet-B5 e eklenen katmanlar gösterilmektedir.



Şekil 3.10. EfficientNet-B5 eklenen katman şeması

Derin öğrenme modellerinin eğitimi sırasında belirli noktalarda çağrılan ve modelin davranışını değiştirmek veya kontrol etmek için kullanılan işlevler vardır. Model eğitilirken kullanılan yöntemler şunlardır:

1. **ModelCheckpoint:** Eğitim safhasında modeli her aşamada kontrol eden bir yöntemdir. Modelin en iyi performans gösterdiği turları kaydetmek için kullanılır. `save_best_only=True` parametresi, sadece en iyi performans gösteren modelin kaydedilmesini sağlamaktadır.
2. **ReduceLROnPlateau:** Bu geri çağırımlar, modelin performansı belirli bir süre boyunca gelişmiyorsa (`patience`) modelin öğrenme hızını azaltır. `Monitor` parametresi, izlenecek metriği belirtir, burada kayıp (`loss`) değeri izlenir. `Factor=0,1` parametresi,

öğrenme hızının azaltılacağı oranı belirtir. Patience=2 parametresi, öğrenme hızının azaltılmasından önce beklenilecek tur sayısını belirtir.

3. EarlyStopping: Bu yöntemde, eğer model belirli bir süre boyunca (patience) belirli bir metrik gelişme göstermiyorsa, eğitimi durdurmaktadır. Monitor parametresi, izlenecek metriği belirtir. Çalışmada doğrulama veri setindeki doğruluk metriği izlenerek modelin eğitiminin durdurulması sağlanmıştır. Patience'ı 5 belirleyerek, eğitimin 5 tur boyunca doğruluk değeri arttırmadığı zaman durdurulması sağlanmıştır.

Model oluşturulduktan ve yöntemler belirlendikten sonra model derleme aşamasına geçer. Derleme aşamasında belirlenmesi gereken birkaç parametre vardır. Aşağıda Çizelge 3.5'te bu parametreler gösterilmektedir.

Çizelge 3.5. EfficientNet-B5 derleme aşamasındaki kullanılan parametreler

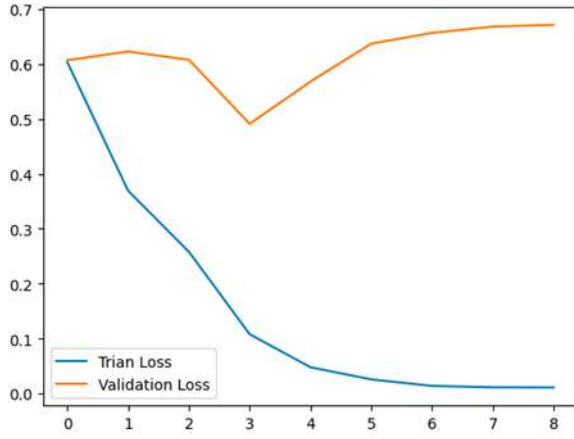
Model	EfficientNet-B5
Optimize edici	Adam
Öğrenme oranı	0,001
Kayıp fonksiyon	Categorical Crossentropy
Metrik	Doğruluk

Modelin compile edilmesi yani derlenmesinden sonra modelin eğitim aşamasına geçilmektedir. Bu aşamada kullanılan parametre ve veri miktarları Çizelge 3.6'da yer almaktadır.

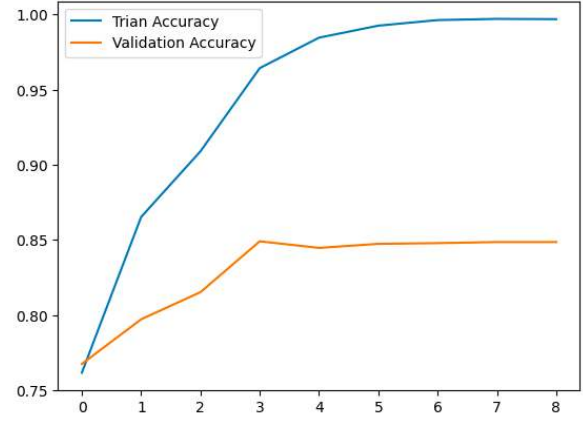
Çizelge 3.6. EfficientNet-B5 eğitim aşamasında parametreler

Eğitim Verisi (Train)	(16414, 224, 224, 3) (16414, 5)
Yığın boyutu	128
Eğitim sayısı	30
Test Verisi (Validation)	(4104, 224, 224, 3) (4104, 5)
Yöntemler	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

Transfer Learning metodu kullanılarak eğitilen EfficientNet-B5 modelinin doğruluk ve kayıp değerlerini gösteren grafikler Şekil 3.11 ve 3.12'de verilmiştir.



Şekil 3.11. EfficientNet-B5 doğruluk değerleri



Şekil 3.12. EfficientNet-B5 kayıp değerleri

Elde edilen grafiklere göre modelin aşırı öğrenmeye en yakın olduğu yer 8. turdaki performansdır. Çünkü eğitim kaybı değeri en düşük değere (0,0112) ulaşırken, doğrulama kaybı değeri ise en yüksek sonucuna erişmiştir, dolayısıyla aradaki fark en büyük olup model aşırı öğrenmeye yakındır. 3.turda ise eğitim ve doğrulama değerlerinin yakın olması aşırı öğrenmeye en uzak olduğu an anlamına gelmektedir. Ayrıca model 3.turda en yüksek doğrulama doğruluk değerine (0,8491) ulaşmış, bundan sonraki 5 turda da gelişme göstermeyince 8. turda eğitim durdurulmuştur.

Ulaşılan son doğruluk değeri 0,84 olup kayıp değeri 0,49'a kadar düşmüştür. Toplamda 5 sınıf için yapılan testlere göre kesinlik, duyarlılık ve f-1-skor değerleri Çizelge 3.7'de gösterilmektedir.

Çizelge 3.7. EfficientNet-B5 metrik tablosu

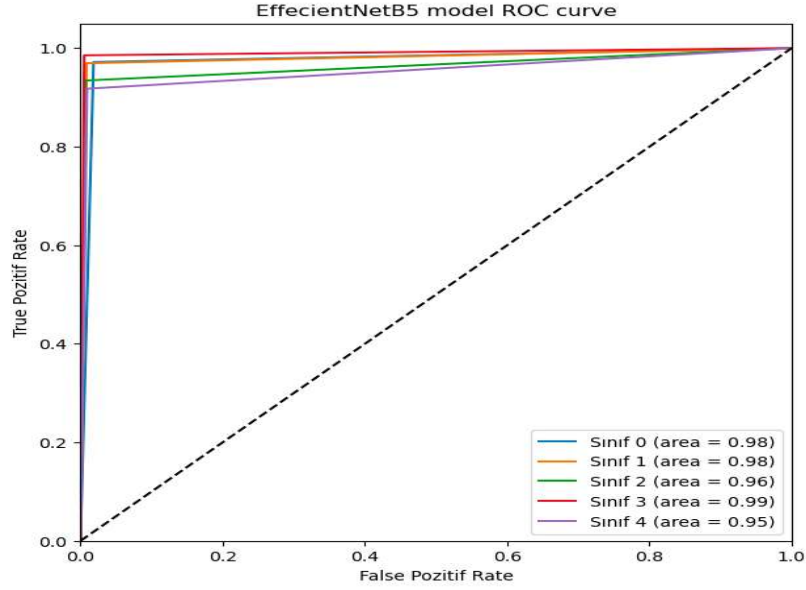
Doğruluk	0,8491	Kayıp	0,4909
	Kesinlik	Duyarlılık	F1-score
Sürpriz	0,79	0,86	0,82
Doğal	0,91	0,90	0,90
Üzgün	0,75	0,73	0,74
Mutlu	0,93	0,94	0,93
Sinirli	0,78	0,73	0,75
Makro ortalama	0,83	0,83	0,83
Ağırlık ortalama	0,85	0,85	0,85

Bu veriler haricinde her bir sınıftan ne kadar sayıda doğru sınıfa atama yapıldığını ve yanlış atamaların hangi sınıflara yapıldığını aşağıdaki Şekil 3.13'te görülmektedir.

surprise	668	28	44	8	32
neutral	26	941	7	61	11
sad	77	6	457	0	85
happy	8	56	2	967	0
anger	65	7	96	0	452
	surprise	neutral	sad	happy	anger

Şekil 3.13. EfficientNet-B5 karmaşıklık matrisi

Sınıflandırma modelinin hassasiyetini ve özgüllüğünü görsel olarak temsil ettiği için ROC eğrisi grafiği oluşturulmuştur. Grafikteki ROC eğrisi her bir duygu sınıfı için modelin doğruluğunu ve hassasiyetini görsel olarak göstermektedir. Y ekseninde duyarlılık (true positive rate) ve X ekseninde özgüllük (false positive rate) yer almaktadır. Bu, modelin doğru pozitifleri ve yanlış pozitifleri nasıl ayırt ettiğini göstermektedir. Bu şekilde her bir duygu sınıfı için ne kadar doğrulukta sınıflandırma işlemi yapıldığı tespit edilebilmektedir. Ayrıca ROC eğrisi, farklı eşik (threshold) değerlerinde modelin duyarlılık ve özgüllük performansını göstermektedir. Optimum eşik değerleri, modelin her duygu sınıfını en iyi şekilde ayırt ettiği noktaların belirlenmesine yardımcı olur. Eğrinin şekli ve eğri altındaki alan (AUC) değeri, modelin genel kararlılığını ve sınıflandırma performansını ölçer. Daha yüksek bir AUC değeri, modelin daha iyi bir sınıflandırma performansına sahip olduğunu gösterir. EfficientNetB5 adlı modelin performansını değerlendirmek için oluşturulan ROC eğrisi grafiği Şekil 3.14'te görülmektedir.

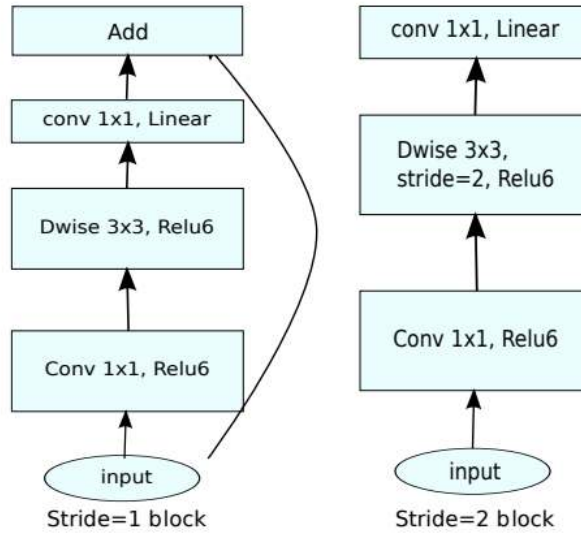


Şekil 3.14. EfficientNet-B5 ROC eğrisi

3.6.3. MobileNetV2

MobileNetV2, derin öğrenme alanında kullanılan bir tür evrişimli sinir ağıdır ve genellikle görüntü sınıflandırma ve nesne tanıma gibi görevlerde tercih edilmektedir. Özellikle MATLAB'da kullanılan MobileNet-v2, ImageNet veri tabanında eğitilmiş olup binlerce nesneyi sınıflandırabilen bir modeldir. Bu ağ, 224x224 piksel boyutundaki görüntüler üzerindeki detaylı ve karmaşık özellikleri öğrenmiştir. Ayrıca, yeni görüntüleri sınıflandırmak veya ağı yeni bir sınıflandırma görevi için yeniden eğitmek için de kullanılabilir.

MobileNetV2 ayrıca derin öğrenme modellerinin küçük boyutlu olması ve düşük hesaplama gücü gereksinimleri nedeniyle mobil cihazlarda ve gömülü sistemlerde yaygın olarak kullanılmaktadır. Bu sayede mobil uygulamalar ve cihazlar üzerinde hızlı ve etkili bir şekilde nesne tanıma ve sınıflandırma işlemleri gerçekleştirilebilmektedir. MobileNetV2'nin şeması aşağıdaki şekilde gösterilmektedir (*MobileNet-v2 convolutional neural network - MATLAB mobilenetv2*, 2024; (Sandler vd., 2019). Şekil 3.15'te MobileNetV2 şeması yer almaktadır.

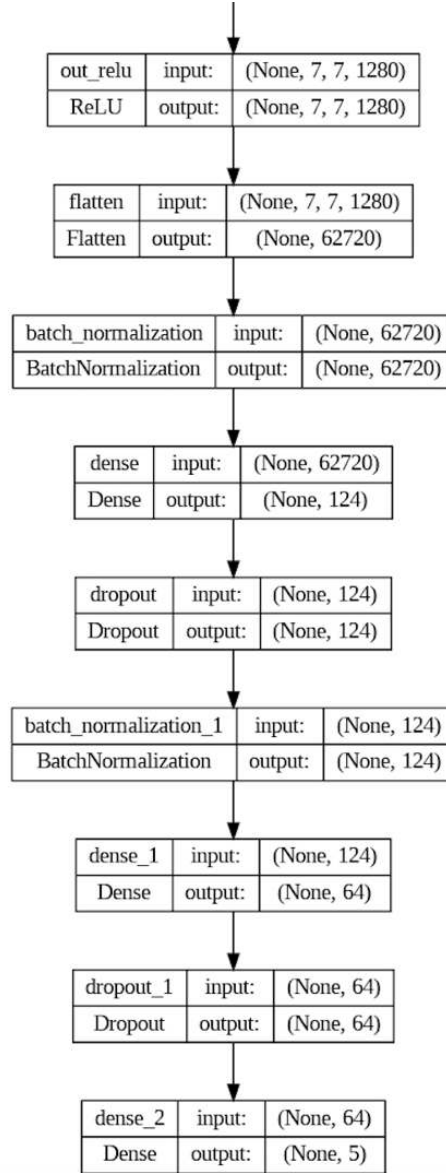


(d) Mobilenet V2

Şekil 3.15. MobilenetV2 model şeması

Uygulamada kullanılan ve temel model olarak alınan modelin üzerine eklenen katmanlar aşağıdaki gibidir:

1. Temel modelin çıktısı alınır ve "include_top=False" parametresi ile üst katmanlar dâhil edilmeden sadece özellik çıkarıcı kısmı alınır.
2. Çıktı düzleştirme işlemi ile düzleştirilir.
3. Düzleştirilmiş çıktıya normalizasyon uygulanır.
4. Düzleştirilmiş ve normalleştirilmiş çıktı, ReLU aktivasyon fonksiyonu ile Dense(124) tam bağlantılı bir katmana iletilir.
5. Ağdaki aşırı uyumu azaltmak için Dropout(0,7) uygulanır.
6. BatchNormalization() ile tekrar normalleştirme uygulanır.
7. Daha sonra, ReLU aktivasyon fonksiyonu ile bir Dense(64) tam bağlantılı katman eklenir.
8. Aşırı uyumu azaltmak için tekrar Dropout(0,4) uygulanır.
9. Son olarak, softmax aktivasyon fonksiyonu ile bir Dense(5) çıktı katmanı eklenir, burada 5 farklı sınıfın olasılıklarını tahmin etmek için kullanılır. Şekil 3.16'da eklenen katmanlar gösterilmektedir.



Şekil 3.16. MobilenetV2 eklenen katman şeması

Eğitim sürecinde kullanılan parametreler, yukarıda açıklanan iki modelle aynı şekilde belirlenmiştir. Model oluşturulduktan ve yöntemler belirlendikten sonra, model derleme aşamasına geçilmektedir. Modelin derlenmesi sırasında belirlenmesi gereken bazı önemli parametreler bulunmaktadır. Çizelge 3.8’de MobilenetV2 modelinin derlenme aşamasındaki parametreler gösterilmektedir.

Çizelge 3.8. MobilenetV2 derleme aşamasında kullanılan parametreler

Model	MobilNetV2
Optimize edici	Adam
Öğrenme oranı	0,001
Kayıp fonksiyonu	Categorical Crossentropy
Metrik	Doğruluk

Modelin derlenmesinden sonra eğitim aşamasına geçilir. Bu aşamada kullanılan parametreler ve veri miktarları Çizelge 3.9’da detaylı bir şekilde belirtilmiştir:

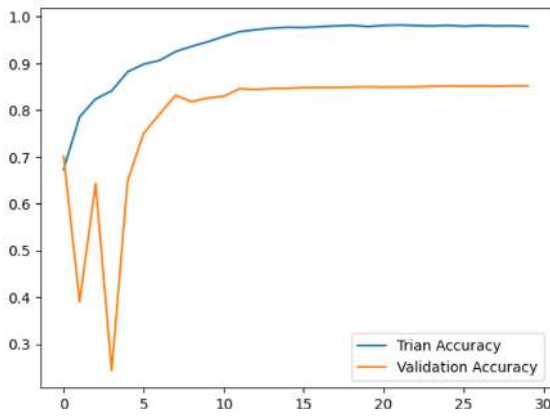
Çizelge 3.9. MobilenetV2 eğitim aşamasında kullanılan parametreler

Eğitim Verisi (Train)	(16414, 224, 224, 3) (16414, 5)
Yığın boyutu	128
Eğitim Sayısı	30
Test Verisi (Validation)	(4104, 224, 224, 3) (4104, 5)
Yöntemler	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

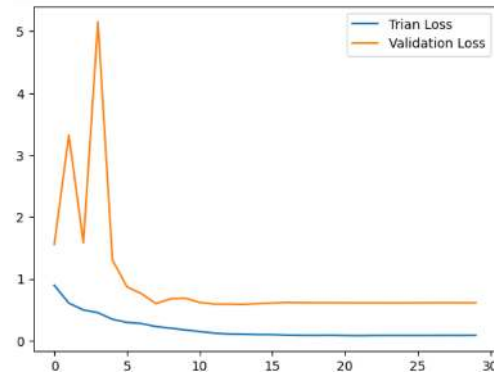
Öğrenme aktarma yöntemiyle eğitilen MobilNetV2 modelinin eğitim sürecinde elde edilen grafikler, hem loss (kayıp) metriği hem de accuracy (doğruluk) metriği üzerinden modelin performansını değerlendirmektedir. Kayıp metriği, modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu ölçmektedir. Bu grafikteki düşüş, modelin eğitim sürecinde kayıpların nasıl azaldığını ve böylelikle modelin öğrenme yeteneğinin arttığını göstermektedir.

Öte yandan, doğruluk metriği modelin verilen veri kümesinde ne kadar doğru tahminler yaptığını belirtmektedir. Eğitim grafikleri aracılığıyla modelin doğruluk metriğindeki değişimleri izlenebilmektedir. ROC (Receiver Operating Characteristic) eğrisi ise modelin sınıflandırma performansını değerlendirmek için kullanılır ve modelin farklı sınıflar arasındaki ayırım yeteneğini gösterir.

Bu grafikler sayesinde modelin eğitim sürecini, performansını ve genelleme yeteneğini daha detaylı bir şekilde analiz etmek mümkündür. Şekil 3.17 ve 3.18’de modelin doğruluk ve kayıp grafiği gösterilmektedir.



Şekil 3.17. MobilenetV2 doğruluk değerleri



Şekil 3.18. MobilenetV2 kayıp değerleri

Yukarıdaki grafikte MobilnetV2 modelinde elde edilen doğruluk ve kayıp değerlerine erişilmiştir. Eğitim doğruluğu %97,94 doğrulama doğruluğu ise %85 bulunmuştur. Eğitim kaybı 0.0881, doğrulama kaybı ise 0,6126 çıkmıştır. Bu modelde yüksek doğruluk oranına erişilmekle birlikte doğrulama verisinde de yüksek performansa ulaşılmıştır. Fakat doğrulama kaybının eğitim kaybından yüksek olması bir miktar aşırı öğrenme belirtisidir. Model aşırı öğrenmeye en yakın 3. turda yakınlık göstermiştir. Bu sebeple model dışarıdan alınan verilere karşı genelleme yapma yeteneğinin eğitim verilerindeki performansından biraz daha düşük olacağı anlamına gelir. Bu modelde doğrulama başarısı en yüksek 13. turda ulaşılmış olup model en iyi tahminleme ve genelleme yeteneğini burada göstermektedir. Bu noktadan daha yüksek bir doğruluğa erişemeyen model 18. turda erken durdurma haline geçmiştir. ROC eğrisi (Receiver Operating Characteristic curve), bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir grafikdir. ROC eğrisi, duyarlılık ve özgünlük arasındaki ilişkiyi göstermektedir. Bir sınıflandırma modeli, bir eşiği (threshold) belirleyerek bir girdiyi pozitif veya negatif olarak sınıflandırır. ROC eğrisi, bu eşik değerini değiştirerek farklı duyarlılık ve özgünlük değerleri elde eder. Veri setindeki her bir sınıf için ROC eğrisi, o sınıfa ait pozitif ve negatif örneklerin tahminlerini değerlendirir. 5 farklı sınıf için ROC eğrisi şu şekildedir:

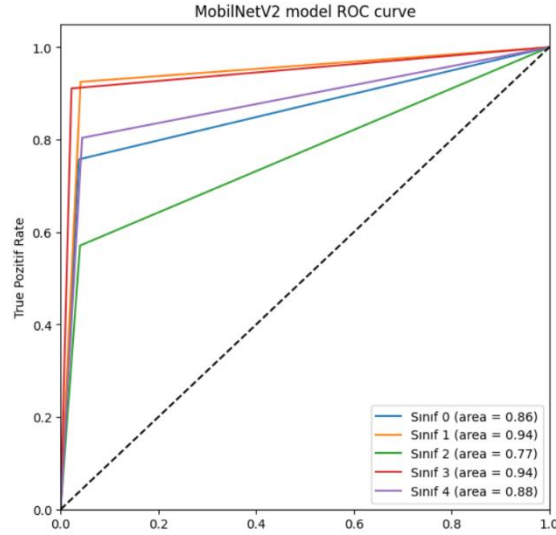
1. Her bir sınıf için ROC eğrisi, o sınıfın pozitif örneklerinin tahminlerini değerlendirirken diğer sınıfların pozitif örneklerini negatif olarak kabul eder.

2. Eğri, farklı eşik değerlerinde duyarlılık ve özgünlük arasındaki ilişkiyi gösterir. Eğri, sol üst köşeye doğru hareket ettiğinde modelin performansı artar.

3. ROC eğrisinin altında kalan alan (AUC), modelin sınıflandırma performansını tek bir sayısal değerde özetler. AUC değeri, 1'e ne kadar yakınsa modelin performansı o kadar iyidir.

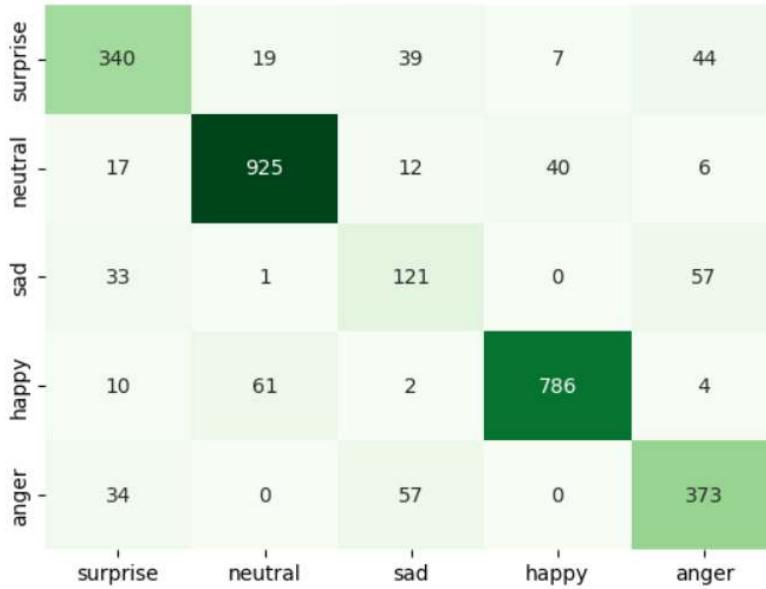
Sonuç olarak her bir sınıf için ROC eğrisi, modelin bu sınıfı doğru bir şekilde sınıflandırma yeteneğini değerlendirir ve AUC değeri, modelin genel performansını ölçmek için kullanılır.

Aşağıdaki Şekil 3.19'da her bir sınıf için ROC eğrisindeki değerler gösterilmektedir.



Şekil 3.19. MobilenetV2 ROC eğrisi

Model eğitildikten ve test edildikten sonra ortaya çıkan karmaşıklık matrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılmıştır. Bu grafik gerçek ve tahmin edilen sınıflar arasındaki ilişkiyi göstermektedir. Bu matris, Gerçek Pozitif (TP), Yanlış Pozitif (FP), Yanlış Negatif (FN) ve Gerçek Negatif (TN) gibi öğeleri içermektedir. Modelin sınıfları doğru ayırıp ayırmadığı, yanlış sınıflandırmaların türü ve belirli sınıflar için modelin performansı net bir şekilde gösterilmektedir. Aşağıdaki Şekil 3.20’de bu değerler her bir sınıf için gösterilmektedir:



Şekil 3.20. MobilenetV2 karmaşıklık matrisi

MobilNetV2 modeli için son doğruluk değeri 0,85 olup kayıp değeri 0,61'e kadar düşmüştür. Toplamda 5 sınıf için yapılan testlere göre kesinlik, duyarlılık ve f-1-skor değerleri Çizelge 3.10'da gösterilmektedir:

Çizelge 3.10. MobilenetV2 metrik tablosu

Doğruluk	0,85	Kayıp	0,61
	Kesinlik	Duyarlılık	F1-score
Sürpriz	0,78	0,76	0,77
Doğal	0,92	0,93	0,92
Üzgün	0,52	0,57	0,55
Mutlu	0,94	0,93	0,93
Sinirli	0,77	0,79	0,79
Makro Ortalama	0,79	0,79	0,79
Ağırlık Ortalama	0,85	0,85	0,85

3.6.4. ResNet50

ResNet-50, Residual Network olarak da bilinen ve 2015 yılında Zhang Xiangyu, Ren Shaoqing ve Sun Jian tarafından tanıtılan bir tür evrişimli sinir ağıdır. ResNet-50; 48 evrişim katmanı, bir MaxPool katmanı ve bir ortalama havuz katmanından oluşan 50 katmanlık bir evrişimli sinir ağıdır. ResNet-50 modeli, geleneksel sinir ağlarına (örneğin VGG-16 ve VGG-19) kıyasla daha az karmaşıklığa sahip olup gradyan kaybolma sorunuyla karşılaşmadan daha fazla evrişim katmanı eklemeyi mümkün kılan kısayol bağlantıları konseptini kullanmaktadır. Bu model, görüntü sınıflandırma ve nesne tanıma gibi görevlerde kullanılmak üzere tasarlanmıştır (Boesch, 2023).

Bu uygulama için seçilen ResNet50'nin eğitim aşamasındaki girdi verileri aşağıdaki gibidir:

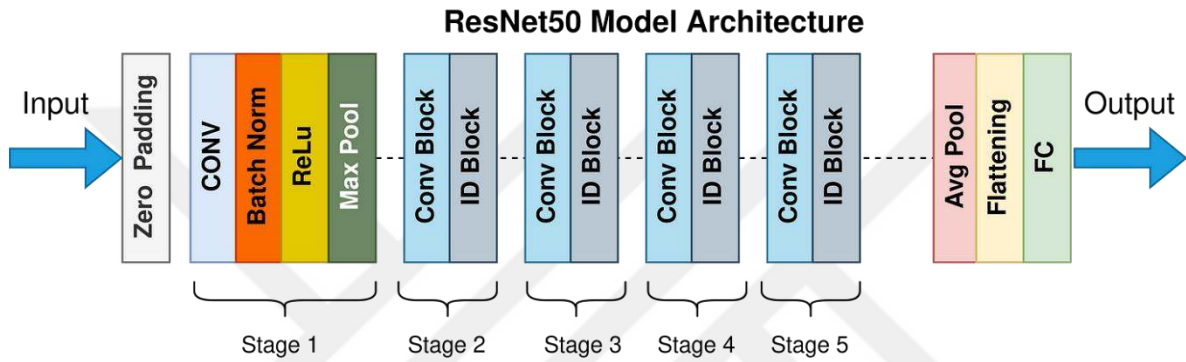
1. Giriş şekli (input shape): Görsellerin boyutu ve RGB renk uzayında (224, 224, 3) olarak belirlenmiştir. Bu, 224x224 piksel boyutunda ve 3 kanallı (RGB) bir giriş şekli olduğundan verideki resimlere uygun olarak belirlenmiştir.

2. Ağırlıklar (weights): Modelin ağırlıkları, ImageNet veri kümesinde önceden eğitilmiş ağırlıklar olarak belirlenmiştir. Böylece, transfer öğrenme yöntemi kullanılarak modelin önceden eğitilmiş ağırlıkları aktarılmıştır.

3. `include_top`: Bu parametreyi *False* olarak değiştirmek, modelin üst katmanlarını (tam bağlantı katmanları) dâhil etmeme seçeneği sunarak resimlerden özellik çıkarmak için modeli bu problem için kullanmayı sağlamıştır.

4. `base_model.trainable`: Bu parametre *True* olarak ayarlandığından, modelin eğitim sırasında ağırlıklarının güncellenmesine izin verilerek modelin eğitildikçe ağırlıklarının veriye uyumunu sağlamıştır.

ResNet-50 modeli, 50 katmanlı bir evrişimli sinir ağıdır ve aşağıdaki şekilde katman yapısı gösterilmektedir (“Keras ResNet50”, 2023). Şekil 3.21’de ResNet50 modelinin şeması verilmiştir.



Şekil 3.21. ResNet50 model şeması

ResNet50’nin temelde 7 farklı katmanı vardır. Bu katmanlar şunlardır:

1. Evrişim Katmanları:

- Evrişim Katmanları, evrişimli sinir ağlarının temel yapı taşlarından biridir.
- 7x7 çekirdek boyutuna sahip olup 64 evrişim katmanı, 64 farklı özellik haritasını öğrenmeye çalışan 64 adet evrişim filtresi içerir.
- Her adımda (stride) filtreler görüntüde 2 piksel kayar.
- Havuzlama katmanı, evrişimli katmanlardan elde edilen özellik haritalarını küçültmek ve önemli özellikleri vurgulamak için kullanılır.
- Havuzlama Katmanında, her 2x2 piksel bloğundan en büyük değeri seçerek bir alt örnekleme işlemi gerçekleştirilir.

2. Bottleneck tasarım katmanı:

3 adet katman içeren bottleneck tasarımı kullanılır. Bu tasarım, parametre sayısını ve matris çarpımlarını azaltarak daha hızlı eğitim sağlar.

3. Tekrarlı katmanlar:

3 defa tekrarlanan; 1x1 çekirdek boyutuna sahip 64 adet, ardından 1x1 çekirdek boyutuna sahip 64 adet ve son olarak 1x1 çekirdek boyutuna sahip 256 evrişim katmanından oluşur. Bu 3 katmanlık blok toplamda 3 kez tekrarlanır.

4. Daha fazla tekrarlı katman:

Toplamda 4 kez tekrarlanan; sırasıyla 1x1 çekirdek boyutuna sahip 128 adet, ardından 3x3 çekirdek boyutuna sahip 128 adet ve son olarak 1x1 çekirdek boyutuna sahip 512 evrişim katmanından oluşur

5. Çok daha fazla tekrarlı katman:

Toplamda 6 kez tekrarlanan; sırasıyla 1x1 çekirdek boyutuna sahip 256 adet, ardından sırasıyla 2x2 ve 3x3 çekirdek boyutuna sahip 256 adet ve son olarak 1x1 çekirdek boyutuna sahip 1024 evrişim katmanından oluşur

6. Final katmanları:

Toplamda 3 kez tekrarlanan; sırasıyla 1x1 çekirdek boyutuna sahip 512 adet, ardından sırasıyla 3x3 çekirdek boyutuna sahip 512 adet ve son olarak 1x1 çekirdek boyutuna sahip 2048 adet evrişim katmanları. Bu noktada ağı toplamda 50 katmana sahip olduğu belirtilmektedir.

7. Çıktı katmanı:

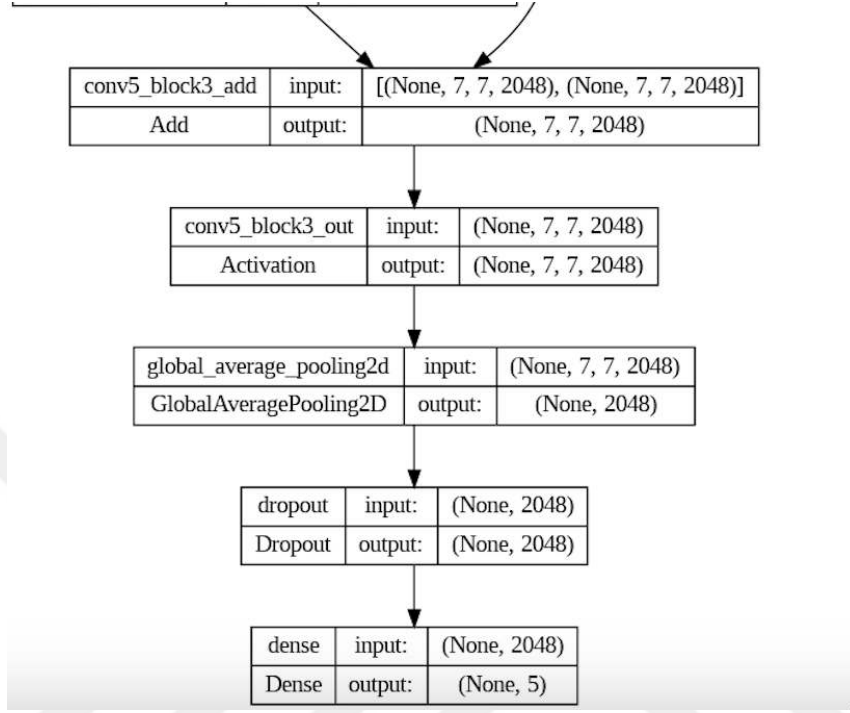
Ortalama havuzlama (average pooling) katmanı ve ardından softmax aktivasyon fonksiyonu ile donatılmış tam bağlantılı bir katman bulunur (He, Zhang, Ren ve Sun, 2015).

Öğrenme Aktarma ile ResNet50'ye eklenen katmanlar şu şekilde eklenir:

1. İlk olarak, ResNet50 modeli önceden eğitilmiş ağırlıklarla yüklenir. Bu ağırlıklar, genellikle ImageNet veri kümesi üzerinde eğitilmiş olan genel özellikleri içerir.
2. Ardından, bu modelin üzerine eklenmek üzere yeni katmanlar oluşturulur. Bu katmanlar, mevcut özellik çıkarma katmanlarının (include_top=False ile belirtilen) çıkışı üzerine eklenir.
3. GlobalAveragePooling2D katmanı eklenir. Bu katman, her özellik haritası kanalı için ortalama değeri olarak özellik haritasını küçültür ve modelin genelleme yeteneğini artırır.
4. Modelin aşırı öğrenme sorununu azaltmak için seyreltme katmanı eklenir. Seyreltme rastgele birimleri devre dışı bırakarak ağı genelleme yeteneğini artırır.
5. Son olarak, çıkış katmanı eklenir. Bu katman, verilen sınıf sayısına göre (burada 5 sınıf için softmax aktivasyonu) çıkış yapar ve modelin duygusal ifadeleri tanımasını sağlar. Bu adımlar, önceden eğitilmiş ResNet50 modeline eklenen ek katmanlar aracılığıyla transfer öğrenmenin nasıl gerçekleştirileceğini gösterir. Bu şekilde, mevcut bir modelin

özelliklerini kullanarak daha az veriyle yeni bir görev için model eğitilebilir ve öğrenme süreci hızlandırılabilir.

Bu katmanların ResNet50 modeline eklendikten sonra çıkan şema aşağıda Şekil 3.22’de gösterilmektedir:



Şekil 3.22. ResNet50 eklenen model şeması

Eğitim sırasında kullanılan parametreler yukarıda açıklanan iki model ile aynı şekildedir.

Model oluşturulduktan ve yöntemler belirlendikten sonra model derleme aşamasına gelir. Derleme aşamasında belirlenmesi gereken birkaç parametre vardır. Aşağıdaki Çizelge 3.11’de ResNet50’ye ait bu parametreler gösterilmektedir:

Çizelge 3.11. ResNet50 derleme aşamasında kullanılan parametreler

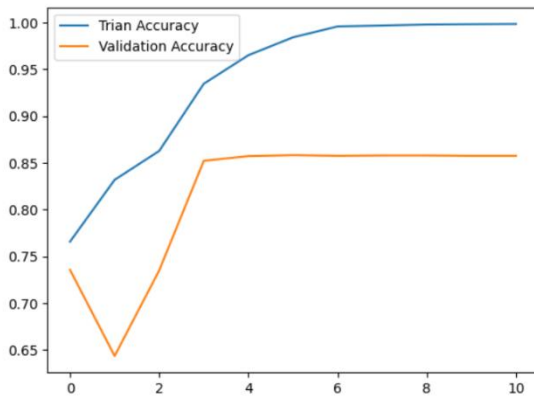
Model	ResNet50
Optimize Edici	Adam
Öğrenme Oranı	0,001
Kayıp Fonksiyon	Categorical Crossentropy
Metrik	Doğruluk

Modelin compile edilmesi yani derlenmesinden sonra modelin eğitim aşamasına geçilir. Bu aşamada kullanılan parametre ve veri miktarları Çizelge 3.12’de gösterilmektedir.

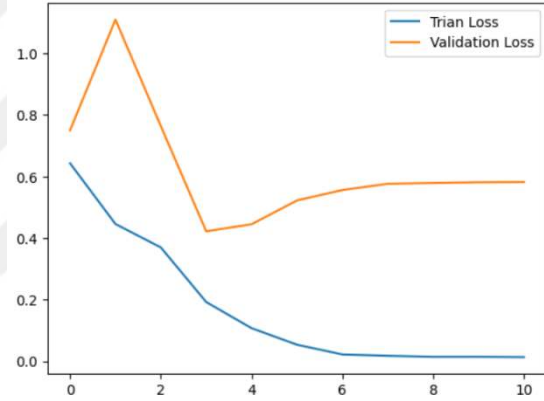
Çizelge 3.12. ResNet50 eğitim aşamasında kullanılan parametreler

Eğitim Verisi (Train)	(16414, 224, 224, 3) (16414, 5)
Yığın boyutu	128
Eğitim sayısı	30
Test Verisi (Validation)	(4104, 224, 224, 3) (4104, 5)
Yöntemler	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

Transfer Learning metodu kullanarak eğitilen ResNet50 modelinin eğitim grafikleri Şekil 3.23 ve Şekil 3.24’te gösterilmektedir.



Şekil 3.23. ResNet50 doğruluk grafiği



Şekil 3.24. ResNet50 Kayıp Grafiği

Ulaştığımız son doğruluk değeri 0,8580 olup kayıp değeri 0,5222’ye kadar düşmüştür. Model eğitim süresince 1. çalıştırma anında %76’lık bir performans sergilerken 11. turda %99,82 ye yakın bir eğitim doğruluk sonucu elde edilmiştir. En iyi başarıyı ise 6. turda sağlamıştır. Bu noktada eğitim doğruluğu %98 çıkarken doğrulama değeri ise %85 tir. Eğitim kaybında ise kaybın 0,6429’dan 0,134’e kadar düşmesi bu modelin veri seti üzerinde mükemmel bir sonuç elde edildiğinin göstergesidir. Model 11. turda daha fazla iyileşme sağlayamadığından dolayı erken durdurma haline gelmiştir. Doğrulama doğruluğu üzerinde de %85 civarında bir başarıyı elde edilmiş, doğrulama kaybının ise 0,5222’ye kadar düştüğü gözlemlenmiştir. Modelin çalışmaya başladığı ilk turda doğrulama kaybı ile eğitim kaybının arasındaki farkın artmasından dolayı model aşırı öğrenme riski taşımış, erken durdurma ve öğrenme oranının azaltılmasıyla bu riskin önüne geçilmiştir. Ayrıca model 11. adımda

doğrulama kaybının daha fazla iyileşme göstermediğini tespit etmesinden dolayı erken durdurma haline geçmiştir. Toplamda 5 sınıf için yapılan testlere göre kesinlik, duyarlılık ve f-1-skor değerleri Çizelge 3.13'te gösterilmektedir:

Çizelge 3.13. ResNet50 metrik tablosu

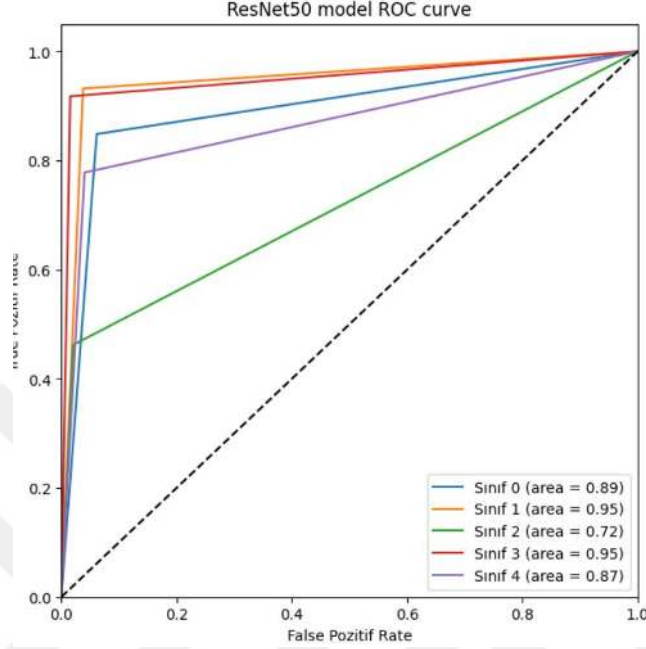
Doğruluk	0,8580	Kayıp	0,5222
	Kesinlik	Duyarlılık	F1-score
Sürpriz	0,71	0,83	0,77
Doğal	0,93	0,93	0,93
Üzgün	0,63	0,46	0,53
Mutlu	0,96	0,92	0,94
Sinirli	0,78	0,78	0,78
Makro ortalama	0,80	0,79	0,79
Ağırlık ortalama	0,86	0,86	0,86

ResNet-50 modeliyle eğitilen duygusal veri setinin performansını değerlendirmek için karmaşıklık matrisi ve ROC eğrisi gibi metrikler kullanılabilmektedir. Bu metrikler, modelin sınıflandırma yeteneğini anlamak için önemli araçlardır. Karmaşıklık matrisi, modelin her bir sınıf için ne kadar doğru ve yanlış tahmin yaptığını gösteren bir matristir. Aşağıdaki şekilde her bir sınıf için modelin kaç tane doğru tahmin yaptığını ve yaptığı yanlışları hangi sınıflarda yaptığını görmek mümkündür. Bu şekil, modelin her bir sınıf için ne kadar etkili olduğunu anlamaya da yardımcı olmaktadır. Ayrıca modelin performansının daha detaylı bir şekilde değerlendirilmesini sağlamaktadır. Şekil 3.25'te ResNet50'ye ait karmaşıklık matrisi verilmiştir.



Şekil 3.25. ResNet50 karmaşıklık matrisi

ResNet50 modelinin çalıştırılması sonucunda her bir sınıf için; modelin sınıfları doğru bir şekilde sınıflandırma yeteneğini, AUC değeri ve modelin genel performansını ölçmek için kullanılan ROC eğrisi aşağıdaki şekilde gösterilmiştir:



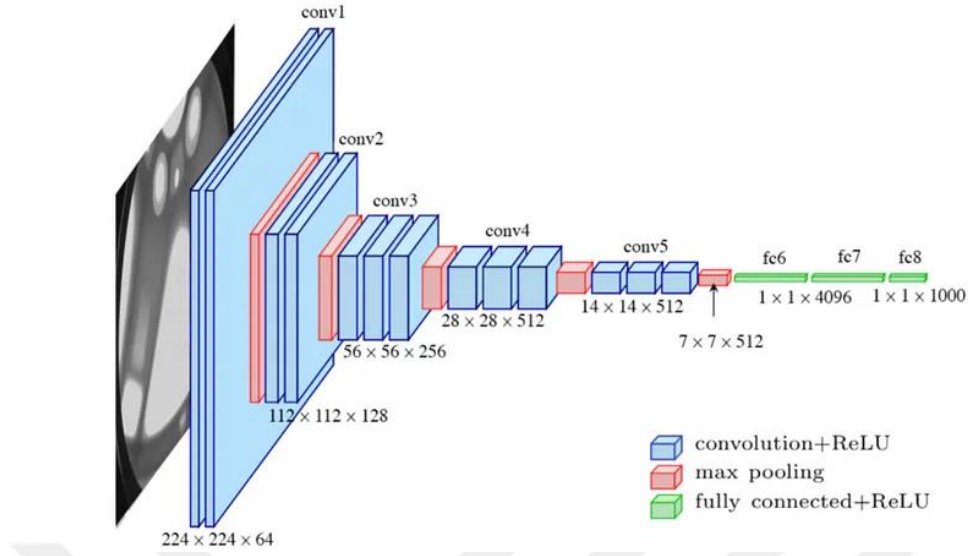
Şekil 3.26. ResNet50 ROC grafiği

3.6.5. VGG16

VGG16, Visual Geometry Group tarafından geliştirilen ve 16 katmandan oluşan bir evrişimli sinir ağı modelidir. Bu model, görüntü tanıma için kullanılan bir derin öğrenme modelidir ve ImageNet veri tabanında yüksek doğruluk elde etmiştir. VGG16'nın temel özellikleri şunlardır:

1. 16 katmanlı bir evrişimli sinir ağıdır ve sadece 3x3 filtre boyutları kullanır.
2. Evrişim ve havuzlama katmanlarının ardışık olarak kullanılmasıyla oluşturulmuş homojen bir yapıya sahiptir.
3. Son katmanlarında iki tam bağlantılı (fully connected) katman ve bir softmax çıkış katmanı bulunur.
4. Toplamda yaklaşık 138 milyon parametreye sahip büyük bir ağıdır.
5. Eğitimi uzun sürebilir ve yüksek hesaplama gücü gerektirebilir.

VGG16 modeli, görüntü sınıflandırma problemlerinde başarılı sonuçlar veren önemli bir derin öğrenme modelidir ve genellikle transfer öğrenme ve ince ayar gibi senaryolarda tercih edilir (Boesch, 2021; Sugata ve Yang, 2017). Şekil 3.27'de VGG16 modelinin şeması verilmiştir.



Şekil 3.27. VGG16 model şeması

Eğitim sırasında kullanılan parametreler yukarıda açıklanan modeller ile aynı şekildedir.

Model oluşturulduktan sonra ve yöntemler belirlendikten sonra model derleme aşamasına gelir. Derleme aşamasında belirlenmesi gereken birkaç parametre vardır. Aşağıdaki Çizelge 3.14’te VGG16 ya ait parametreler gösterilmektedir:

Çizelge 3.14. VGG16 derleme aşamasında kullanılan parametreler

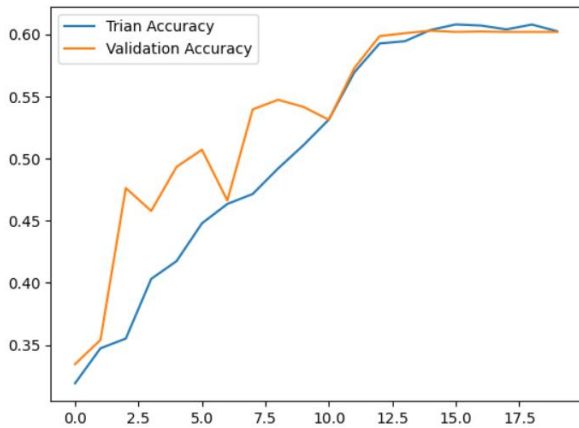
Model	VGG16
Optimize Edici	Adam
Öğrenme Oranı	0,001
Kayıp Fonksiyon	Categorical Crossentropy
Metrik	Doğruluk

Modelin derlenmesinden sonra modelin eğitim aşamasına geçilir. Bu aşamada kullanılan parametre ve veri miktarları Çizelge 3.15’te gösterilmektedir:

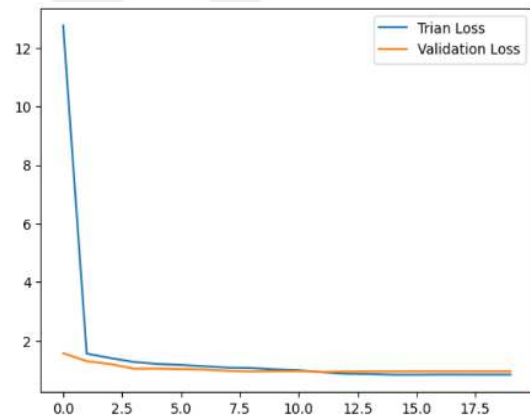
Çizelge 3.15. VGG16 Eğitim aşamasında kullanılan parametreler

Eğitim Verisi (Train)	(16414, 224, 224, 3) (16414, 5)
Yığın boyutu	128
Eğitim sayısı	30
Test Verisi (Validation)	(4104, 224, 224, 3) (4104, 5)
Yöntemler	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

Transfer Learning yöntemiyle eğitilen VGG16 modelinin eğitim grafikleri, loss (kayıp) ve accuracy (doğruluk) metriklerini içerir. Kayıp grafiği, modelin eğitim sürecinde kayıpların nasıl azaldığını gösterirken doğruluk grafiği, modelin doğru tahminlerinin yüzdesini belirtir. Bu model için genel bir değerlendirme yapıldığında eğitim kaybının ilk turda 12,75 gibi yüksek bir değere ulaştığı ancak 20. turda 0,8478'e kadar düşmesi bu modelin giderek daha iyi öğrendiğinin göstergesidir. Eğitim doğruluğu tarafında ise ilk turda %31,62 iken 20. turda %60,23'e ulaşmıştır. Doğrulama doğruluğu ise ilk turda %33'ten %60,17'ye çıkmıştır. Bu sonuçlara bakılarak modelin eğitim verisinde iyi bir performans gösterdiğini söylemek mümkün iken doğrulama verileri için aynı şeyleri söylemek doğru değildir. Aradaki bu fark aşırı öğrenmeyi işaret etse de her 2 metriğin de gelişme kaydetmesi bu durumun kesin olduğunu göstermemektedir. Ancak 11. eğitimden sonraki süreçte dikkat edilmelidir. Yine model 1. turdan 10.tura kadar eksik öğrenme belirtisi göstermektedir. Model en iyi performansına 15. turda erişmiş, bu noktadan daha iyi bir ilerleme kaydedemediğinde 20. turda erken durma yapmıştır. Şekil 3.28 ve Şekil 3.29'da VGG16 modeline ait doğruluk ve kayıp değerleri gösteren grafikler verilmiştir.

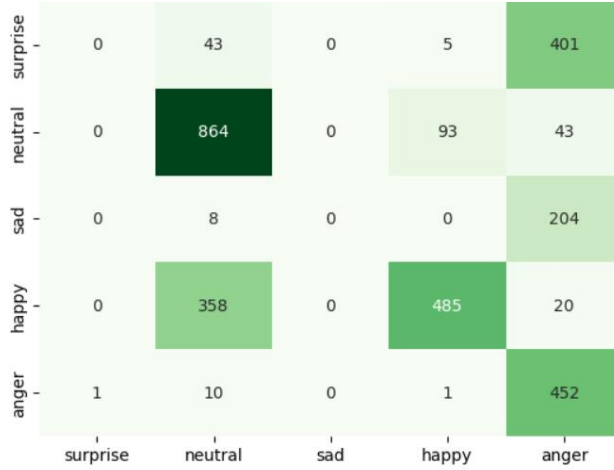


Şekil 3.28. VGG16 doğruluk grafiği

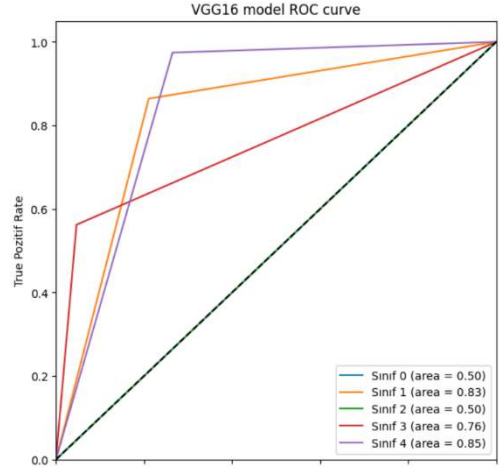


Şekil 3.29. VGG16 kayıp grafiği

Modelin test edilmesi sonucunda çıkan karmaşıklık matrisi ve ROC eğrisi grafikleri Şekil 3.30 ve Şekil 3.31'de verilmiştir:



Şekil 3.30. VGG16 karmaşıklık matrisi



Şekil 3.31. VGG16 ROC eğrisi

Ulaşılan son doğruluk değeri 0,6017 olup kayıp değeri 0,9605'e kadar düşmüştür. Toplamda 5 sınıf için yapılan testlere göre kesinlik, duyarlılık ve f-1-skor değerleri aşağıdaki Çizelge 3.16'da gösterilmektedir:

Çizelge 3.16. VGG16 metrik tablosu

Doğruluk	0,6017	Kayıp	0,9605
	Kesinlik	Duyarlılık	F1-score
Sürpriz	0,00	0,00	0,00
Doğal	0,67	0,86	0,76
Üzgün	0,00	0,00	0,00
Mutlu	0,83	0,56	0,67
Sinirli	0,40	0,97	0,57
Makro ortalama	0,38	0,48	0,40
Ağırlık ortalama	0,53	0,60	0,54

VGG16 modeli, diğer modellere göre daha derin ve daha fazla parametre içeren bir mimariye sahiptir, bu da daha yüksek hesaplama maliyetleri ve aşırı öğrenme problemine yol açabilir. Ayrıca VGG16'da bazı sınıflarda modelin hiç başarı sağlayamadığı görülmüştür.

Daha az karmaşıklık gerektiren senaryolar için daha uygun olan MobileNetV2 ve ResNet50 gibi diğer modeller, daha hafif ve daha verimli mimarilere sahiptir. MobileNetV2, ters residual yapı ile özellik çıkarma yeteneklerini artırırken ResNet50 kaybolan gradyan sorununu çözmek için artık bağlantıları kullanır. MobileNetV2 ayrıca doğrusal olmayanları

kaldırarak modelin karmaşık desenleri daha etkili bir şekilde yakalama yeteneğini artırır. VGG16 bu konuda bu modellere göre bu problem için uygun değildir.

3.6.6. YOLOv8m-cls

2016 yılında Redmon ve diğer araştırmacılar tarafından YOLO (You Only Look Once) adlı nesne tespiti (object detection) alanında kullanılan bir derin öğrenme algoritması geliştirilmiştir. Bu model, Imagenet veri tabanındaki 1000 sınıfa ait görüntülerle eğitilmiş ve 24 evrişimli katman, 2 tamamen bağlı katman, 1 x 1 indirgeme katmanı ve 3 x 3 evrişimli katman içeren bir sinir ağı kullanmıştır. YOLO, gerçek zamanlı olarak saniyede 45 kare işleyebilen ve giriş olarak aldığı görüntüleri 448 x 448 piksel boyutuna yeniden boyutlandırarak evrişimli sinir ağı ile nesne algılama işlemini gerçekleştiren bir yapıya sahiptir. YOLO, nesnelerin genel temsillerini öğrenir ve DPM ve R-CNN gibi diğer algılama yöntemlerinden daha iyi performans gösterir.

YOLO, bir görüntüyü tek bir işlemde analiz ederek nesneleri tespit etmeyi ve sınıflandırmayı sağlar. YOLO algoritması, nesnelerin sınıflarını tahmin etmek için ağın çıkışında bulunan bounding box'ları kullanır. Bu bounding box'lar, nesnelerin konumunu ve boyutunu belirler. YOLO ayrıca her hücre için bir vektör kodlama kullanarak nesneleri tespit eder. Bu vektörler, hücrenin içindeki nesnenin konumunu ve sınıfını temsil eder. YOLOv8-cls ise YOLO algoritmasının geliştirilmiş bir versiyonudur ve daha hızlı ve doğru nesne tespiti yapabilme yeteneğine odaklanmıştır. Bu çalışma mimarisi, doğal görüntülerden sanat eserlerine kadar geniş bir yelpazede nesneleri algılamak için etkili bir şekilde tasarlanmıştır. YOLOv8m-cls, bilgisayarlı görü ve nesne tespiti projelerinde yaygın olarak kullanılmaktadır. Özellikle otonom sürüş, güvenlik, üretimde anormallik tespiti, spor gibi alanlarda başarılı sonuçlar vermektedir. YOLO algoritması ve YOLOv8m-cls modeli, gerçek zamanlı nesne tespiti için yüksek doğruluk ve işleme hızı sağlayarak önemli bir rol oynamaktadır (Zvornicanin, 2024).

YOLOv8m-cls modelini uygulamak ve çalıştırmak için ilk önce veri seti eğitim, test ve doğrulama olarak ayrı klasörlere ayrıldıktan sonra bu klasörlerin bulunduğu ana klasör DATA_DIR değişkeninde tutulmuş ve aşağıdaki komut çalıştırılmıştır:

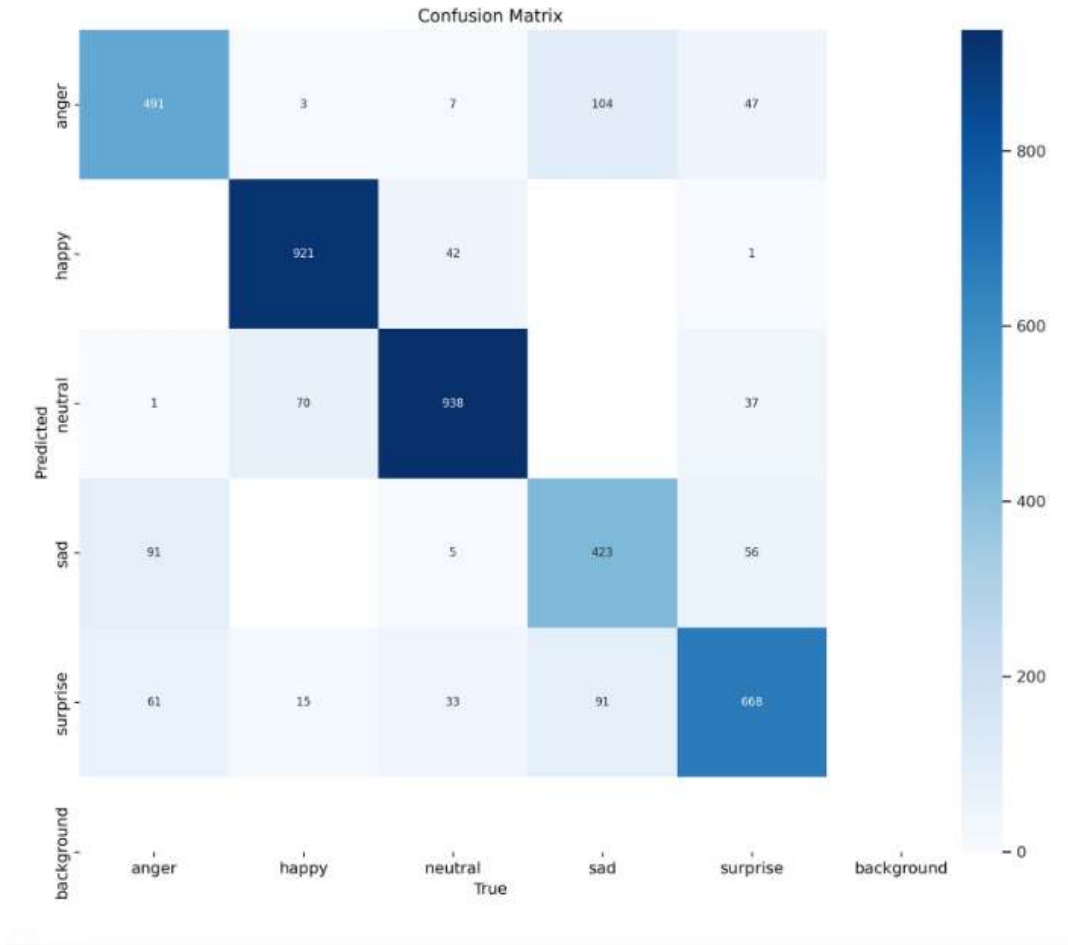
```
!yolo task=classify mode=train model=yolov8m-cls.pt data='{DATA_DIR}' epochs=30  
imgsz=96
```

Bu komut, YOLOv8m-cls modelinin sınıflandırma ve eğitilmesi amacıyla kullanılmaktadır.

1. Komut içindeki "task=classify" parametresi, modelin sınıflandırma göreviyle ilgili olduğunu belirtir.
2. "mode=train" parametresiyle modelin eğitim modunda çalışacağı ifade edilirken, "model=yolov8m-cls.pt" parametresi ise kullanılacak olan önceden eğitilmiş modelin yolov8m-cls.pt adında bir dosya olduğunu göstermektedir.
3. Eğitim için kullanılacak verilerin dizini, "data='{DATA_DIR}'" şeklinde belirtilir ve bu parametre verilerin dizinini temsil eder.
4. "epochs=30" parametresi, modelin toplam 30 tur boyunca eğitileceğini gösterirken "imgsz=96" parametresi ise modelin kullanacağı giriş görüntülerinin boyutunu belirtir.

Özetle bu komut, YOLOv8m-cls modelinin sınıflandırma görevi için belirtilen veri seti üzerinde eğitilmesini sağlar ve modelin eğitim süreci ile ilgili temel parametreleri içerir.

Sınıflandırma işlemi sonucu her bir sınıfın hangi oranda tahmin edildiğini gösteren karmaşıklık matrisi Şekil 3.32'de ki gibidir:



Şekil 3.32. YOLOv8m-cls confusion matrix

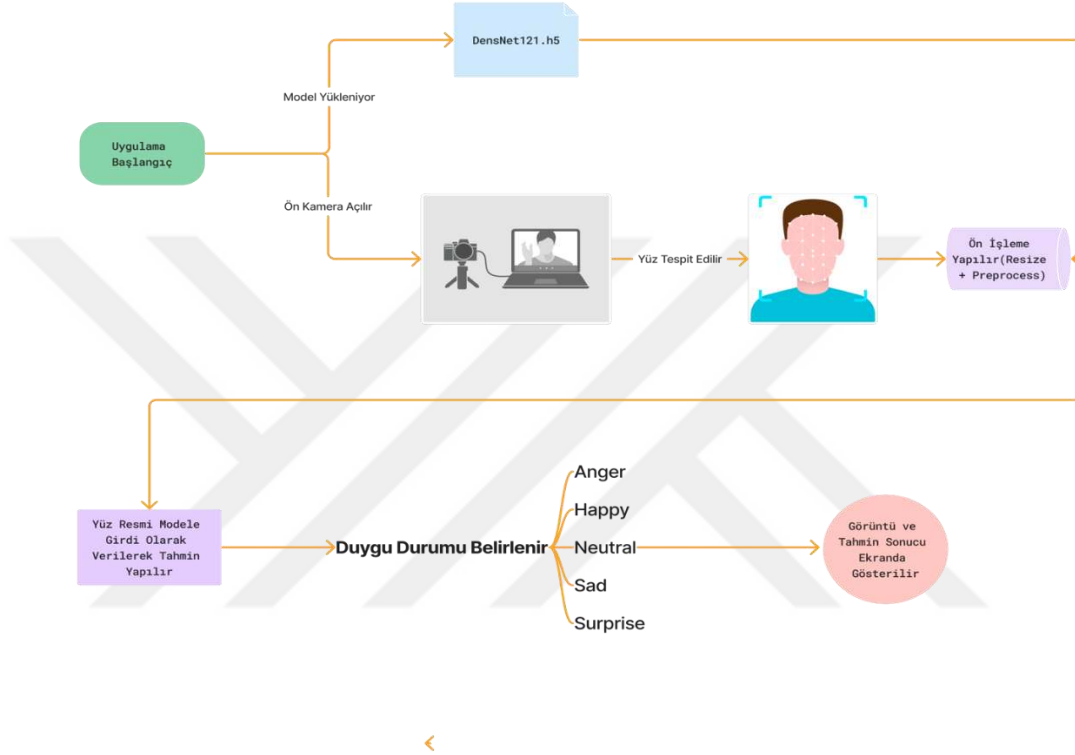
Bu karmaşıklık matrisinden yola çıkarak YOLOv8m-cls modelinin çalışması neticesinde ulaşılan kesinlik, duyarlılık ve f1- skor sonuçları Çizelge 3.17'deki gibidir:

Çizelge 3.17. YOLOv8m-cls değer tablosu

Metrik	Sinirli	Mutlu	Doğal	Üzgün	Sürpriz	Makro ortalama
Doğruluk	0,89	-	-	-	-	0,89
Kesinlik	0,84	0,96	0,96	0,59	0,77	0,84
Duyarlılık	0,87	0,95	0,93	0,88	0,91	0,87
F1 Score	0,84	0,95	0,94	0,71	0,83	0,84

4. BULGULAR

Bu uygulamada duygu durumu tahminlemesi yapmak için bir kullanıcı arayüzü oluşturulmuştur. İşlemler kameradan alınan görüntü üzerinde gerçekleştirilmiştir. Bu uygulamanın çalışma prensibi şekil 4.1'deki gibi gösterilmektedir:



Şekil 4.1. Uygulama çalışma şeması

Uygulamanın daha detaylı çalışma adımları aşağıdaki gibidir:

1. Kivy kütüphanesinden gerekli modüllerin ve sınıfların kütüphaneleri kurulmuştur.
2. Görüntü işleme ve model tahmini için OpenCV, NumPy ve TensorFlow/Keras kütüphanelerini kullanılmıştır.
3. Önceden eğitilmiş bir Densenet121 modeli yüklenmiştir.
4. Duygu durumlarını temsil eden listeler oluşturulmuştur.
5. Kamera bağlantısı açılarak VideoCapture nesnesi oluşturulmuştur (cap = cv2.VideoCapture(0)).
6. Arayüz bileşenleri oluşturularak gösterilmiştir.
7. Uygulama başlatılmıştır.

8. Görüntü güncelleme tanımlanarak 1/30 saniyelik bir zaman aralığı belirlenmiş ve update metodu kullanılarak belirtilen zaman aralığı ile kameradan görüntü karesi alınmıştır.
9. Gelen görüntü modele girdi olarak verildikten sonra görüntü 224x224 piksel boyutlarında yeniden boyutlandırılmış ve RGB formatına dönüştürülmüştür. Sonrasında, model için giriş olarak hazırlanan görüntü, 4 boyutlu bir tensöre dönüştürülerek modele verilmiştir. Model, bu görüntünün sayısal değerlerini kullanarak tahmin gerçekleştirmiş ve çıktı olarak bir duygu sınıfının sayısal karşılığını vermiştir.
10. Çıkan duygu sınıfının sayısal değerinden hangi sınıfa ait olduğu sonucu text olarak çıkarılarak Kivy kütüphanesi yardımıyla ekrana yazdırılmıştır.

Uygulamanın genel amacı, kullanıcıya gerçek zamanlı olarak duygu durumu tahminleme yeteneği sağlamaktır. Kullanıcının görüntüsü alındığında, model tarafından tahmin edilen duygu durumu ekranda gösterilerek kullanıcıya duygu analizi yapma imkânı sunulmaktadır.

Sonuç olarak oluşturulan tablo, farklı görüntü işleme modellerinin performansını göstermektedir. Her model için doğruluk, kayıp, kesinlik, duyarlılık ve F1-skoru ölçümleri yapılmıştır. Doğruluk, modelin genel başarı oranını gösterirken, kayıp değeri tahminler ile gerçek değerler arasındaki hata miktarını ifade etmektedir. Kesinlik, pozitif tahminlerin gerçek pozitiflere oranını, duyarlılık ise gerçek pozitiflerin tüm pozitifler içindeki oranını gösterir. F1-skoru ise kesinlik ve duyarlılığı dengeli bir şekilde değerlendirerek modelin genel performansını ölçer. Tablodaki değerler, her bir modelin duygu tespit veri setinde ve görevinde ne kadar etkili olduğunu göstermektedir. Çizelge 4.1’de tüm modellerin karşılaştırılması verilmiştir.

Çizelge 4.1. Karşılaştırmalı sonuç değer tablosu

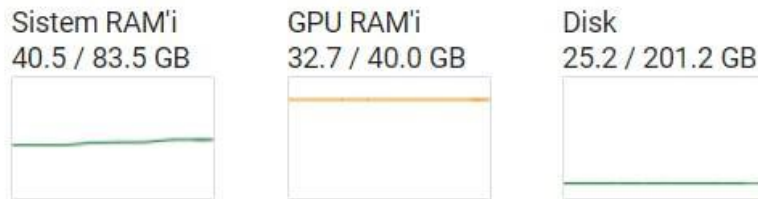
Model	Doğruluk	Kayıp	Kesinlik	Duyarlılık	F1-Score
DenseNet121	0,79	0,57	0,82	0,68	0,64
EfficientNet	0,84	0,49	0,82	0,81	0,82
MobilNetV2	0,85	0,61	0,79	0,79	0,79
ResNet50	0,85	0,52	0,80	0,79	0,79
VGG16	0,60	0,96	0,38	0,48	0,40
YoloV8m-cls	0,89	0,32	0,84	0,87	0,84

Bu sonuçlara göre, farklı metrikler kullanılarak değerlendirilen çeşitli modellerin performansı karşılaştırılabilmektedir. Test edilen modeller arasında DenseNet121, EfficientNet, MobilNetV2, ResNet50 ve Yolov8m-cls 0,89 ile 0,85 arasında değişen daha yüksek doğruluk puanları sergilendiği görülmektedir. Ancak kayıp, hassasiyet, duyarlılık ve F1-skoru gibi diğer metrikler dikkate alındığında Yolov8m-cls ve EfficientNet daha dikkat çekici sonuçlar göstermiş ve çeşitli ölçümlerde tutarlı puanlar elde etmiştir. YOLOv8m-cls modeli, 0,84'lük bir F1-skoru ile hassasiyet ve duyarlılık arasında dengeli bir performans sergilemiştir. Öte yandan, VGG16, diğer modellere göre daha düşük bir doğruluk olan 0,60 ile daha yüksek kayıp ve daha düşük hassasiyet, duyarlılık ve F1-skoru değerleri sergilemiştir. Genel olarak YoloV8 modelinin çeşitli değerlendirme metriklerindeki performanslarına dayanarak en umut verici modeller olduğu sonucuna varılmıştır.

Modellerin GPU ve CPU Kullanımları

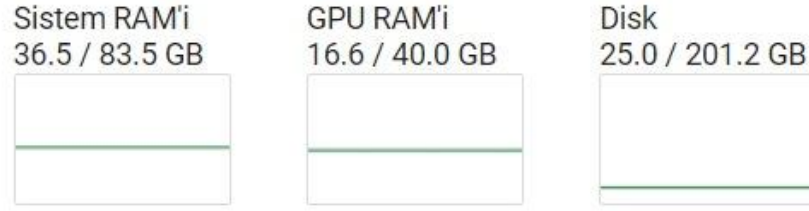
Aşağıdaki grafikler, Google Colab platformunda eğitilen farklı modellerin GPU ve sistem RAM kullanımlarını göstermektedir. Bu kullanımlar, modellerin işlemci ve grafik işlemci kaynaklarının nasıl kullanıldığının daha iyi anlaşılmasına yardımcı olmuştur. İlgili performans metriklerine ek olarak, bu kullanımların verimliliği ve uygulama üzerindeki etkileri de değerlendirilmiştir. Grafikler, modellerin sistem kaynaklarını ne kadar kullandıkları ile ilgili sonuçlardan bahsetmektedir. Şekil 4.2 EfficientNet GPU ve sistem RAM kullanımını göstermektedir.

Python 3 Google Compute Engine arka uç (GPU)
13:28 ile 13:44 arasındaki kaynaklar gösteriliyor



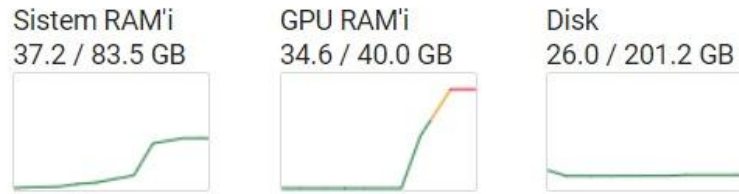
Şekil 4.2. EfficientNet GPU ve sistem RAM kullanımı

Python 3 Google Compute Engine arka uç (GPU)
02:32 itibarıyla kaynaklar gösteriliyor



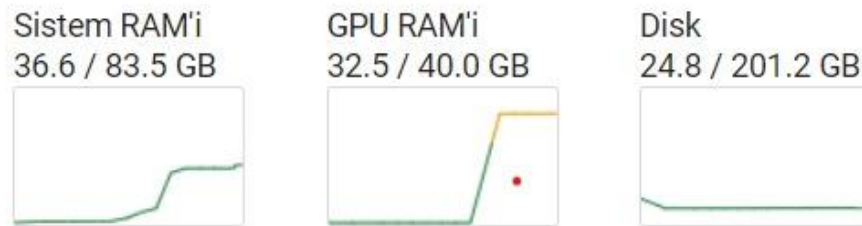
Şekil 4.3. MobileNetV2 modelinin GPU ve RAM kullanımı

Python 3 Google Compute Engine arka uç (GPU)
18:16 ile 18:38 arasındaki kaynaklar gösteriliyor



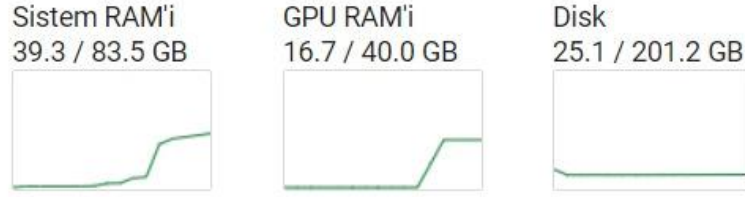
Şekil 4.4. VGG16 GPU ve sistem RAM kullanımı

Python 3 Google Compute Engine arka uç (GPU)
16:13 ile 16:45 arasındaki kaynaklar gösteriliyor



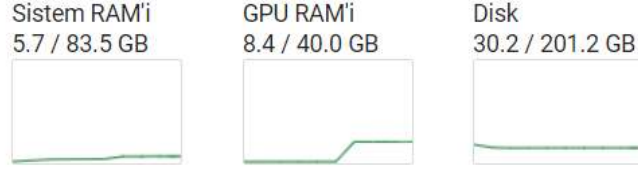
Şekil 4.5. DenseNet GPU ve sistem RAM kullanımı

Python 3 Google Compute Engine arka uç (GPU)
15:05 ile 15:37 arasındaki kaynaklar gösteriliyor



Şekil 4.6. ResNet GPU ve Sistem RAM kullanımı

Python 3 Google Compute Engine arka uç (GPU)
18:35 ile 18:55 arasındaki kaynaklar gösteriliyor



Şekil 4.7. Yolov8m-clc GPU ve sistem RAM kullanımı

Modellerin daha hızlı eğitimi için ve veri setinin büyüklüğünden dolayı eğitimler Google Colab Pro da yapılmıştır. Bu platform bize 83,5 GB Sistem RAM' i, 40 GB GPU RAM ve 201,2 GB disk hizmeti sağlamaktadır. Kaynak kullanımı açısından değerlendirildiğinden en çok sistem ram kullanımı yapan model EfficientNet olurken, GPU rami en çok kullanan model ise VGG16 olmuştur. Tüm kaynakları en az kullanan model ise YoloV8 olduğu sonucu elde edilmiştir. Burada tüm modellerin aynı veri seti üzerinden eğitim yapmasına karşın kaynak kullanımında farklılık olmasının sebebi EfficientNet ve VGG16'nın mimari olarak daha yoğun, karmaşık hesaplamalar içermesi ve model boyutunun daha fazla olmasından dolayıdır. YoloV8 modeli ise diğer modellere göre daha güncel ve hızlı olması nedeniyle görüntü analiz, nesne tespiti, duygu tespiti gibi problemlerde kullanılmaya daha yatkındır.

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında derin öğrenme teknikleri kullanarak duygu analizi sınıflandırılması yapılmıştır. Transfer öğrenme modellerinden DenseNet121, EfficientNetB5, MobilNetV2, ResNet50, VGG16 ve YOLOv8m-cls yöntemleri tercih edilmiştir. Bu modellerin tercih edilmesinin nedenleri daha önceden eğitildikleri için hesaplama maliyeti ve zaman açısından verimlidir. Geniş veri setleri üzerinde eğitilen bu modellerin genelleme kabiliyetlerinin iyi olması sebebiyle bu tez çalışmasında da yüksek başarımlar sağlayacağı düşünülmüştür. Kaggle ortamından alınan 29042 görüntü belirlenen modellerde eğitilmesi amacıyla %80'i eğitim, %20'si test olarak ayrılmıştır. Böylece 16414 görüntü eğitim için, 4104 görüntü ise test için bölünmüştür. Veri setindeki görüntüler 96x96 boyutunda olup 3 renk kanalına (RGB [kırmızı, yeşil, mavi]) sahiptir. Bu görüntüler şaşkın, korku, tarafsız, üzgün, iğrenç, küçümseme, mutlu ve öfke olmak üzere 8 farklı duyguyu içermektedir. Fakat modelin daha iyi bir başarımla elde edebilmesi amacıyla en düşük veriye sahip küçümseme ve iğrenme sınıfları ile korku ve üzüntü sınıflarının da birbirine yakın veriler olmasından dolayı korku sınıfı eğitim başlamadan veri setinden çıkarılmıştır. Modeller eğitim öncesi tüm görüntüler 224x224x3 piksel şeklinde yeniden boyutlandırılmıştır. Bu işlem modelin tahminleme yeteneğinin süresini uzatsa da daha iyi sonuçlar vereceği için yapılmıştır. Transfer öğrenme modellerinde yapılan eğitim sonucunda en yüksek başarımla 0,89 ile YOLOv8m-cls bulunurken sırasıyla Resnet50 için 0,85, MobilNetV2 için 0,85, EfficientNetB5 için 0,84, DenseNet121 için 0,79 ve VGG16 için 0,60 başarımla elde edilmiştir. Modelin eğitim aşamasındaki kayıplarında ise az kaybın 0,32 ile YOLOv8m-cls modelinde, en çok kaybın da yine 0,96 ile VGG16 modelinde olduğunu sonucuna varılmıştır. Genel itibarı ile VGG16 modeli her anlamda bu problem için uygun bir model değilken, YOLOv8m-cls ise bu problem için en uygulanabilir modeldir. Modellerden 5'i en iyi mutluluk duygusunu tahmin etmede başarılı iken, MobilNetV2 ise en iyi nötr duygusunu tahmin etmiştir.

Bu çalışmada görüntüler üzerinden duygu analizi yapılarak kişiler hakkında doğru teşhisi bulmak hedeflenmiştir. Duygu problemi geçmişten günümüze insanları anlamamanın ve doğru bir iletişimin ilk basamağı olup karşılaşılan problemleri çözmek için de anlaşılması önem arz etmektedir. Literatüre bakıldığında da bu alanla ilgili birçok çalışma mevcuttur. Müşteri analizinde, pazarlama ve ürün tanıtımında, sosyal medya analizinde ve sağlık sektöründe benzer çalışmalar bulunmaktadır. Bu çalışmada temel kullanım alanı; okul ve çocuk gelişim merkezlerinde rehberlik hizmeti veren yerler, özel klinikler,

RDM merkezleri veya bu alanla ilgili hizmet veren birimlerdir. Psikologların temel görevi bu kişilerin duygularını doğru tespit edip ona göre tedavi aşamasına geçmektir. Fakat bazı bireylerde özellikle konuşma yetisi olmayan veya başka türlü engelle sahip olan, kendisini açıkça ifade edemeyen bireylerde bunu tespit etmek zor olmaktadır. Derin öğrenme modelleri ile yapılan bu uygulama ile bu problemin önüne geçilmesi planlanmaktadır. Bu alana yönelik araştırmaların az olması bu çalışmanın gerçekleştirilmesinin en önemli sebeplerden biridir. Genellikle duygu analizi işlemleri metinden veya yorum satırlarından elde edilen veri setlerinin kullanımı sonucu bulunan bir işlemdir. Fakat bu çalışmada görüntüler üzerinden bir veri seti tercih edilmiştir. Ayrıca klasik makine öğrenme algoritmaları yerine daha hızlı problem çözen ve yüksek sonuçların elde edildiği transfer öğrenme metodlarının kullanımı farkındalık oluşturmuştur. Klasik duygu analizinde sınıflar genellikle 2 veya 3 sınıfa ayrılırken (pozitif, nötr ve negatif) bu çalışmada daha net anlaşılabilen, duygusal durumu daha iyi açıklayan 5 ayrı sınıf kullanılmıştır. Gelecekte bu çalışmalar üzerine daha büyük veri setleri ile eğitim yapılabilir, veri artırma ve Yolov8m-cls gibi hızlı ve gelişmiş algoritmalar ile doğruluk değerleri daha fazla yükseltilebilir. Çalışmalar mobil uygulamalar hâline getirilirse daha pratik ve kullanışlı hâle gelebilir.

KAYNAKLAR

- Açıkalın, U. U., Bardak, B. ve Kutlu, M. (2020). BERT Modeli ile Türkçe Duygu Analizi. *2021 28th Signal Processing and Communications Applications Conference, SIU 2020-Proceedings*, 1-4. <http://gcris.etu.edu.tr:8080/handle/20.500.11851/4264>
- Akbaş, A. (2021, 16 Ağustos). Yapay Sinir Ağlarında Aktivasyon Fonksiyonları Çeşitleri. *Medium*. 5 Nisan 2024 tarihinde <https://medium.com/kodcular/yapay-sinir-a%C4%9Flar%C4%B1nda-aktivasyon-fonksiyonlar%C4%B1-%C3%A7e%C5%9Fitleri-14c2ab782866> adresinden erişildi.
- Akca, M. F. (2020, 22 Kasım). RNN Nedir? Nasıl Çalışır? *Medium*. 6 Nisan 2024 tarihinde <https://medium.com/deep-learning-turkiye/rnn-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-9e5d572689e1> adresinden erişildi.
- Akdağlı, E. (2021, 11 Nisan). Evrişimsel Sinir Ağları (Convolutional Neural Network). *Medium*. <https://ece-akdagli.medium.com/evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-convolutional-neural-network-a6c9d8e666c4> adresinden erişildi.
- Akdeniz, A. C. (2022). *Makine Öğrenmesi Yöntemleri Kullanılarak Uzaktan Eğitim Konulu Türkçe Tweetlerin Duygu Analizi* (Yayımlanmamış Yüksek Lisans Tezi), Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Konya.
- Akköse, O. (2021, 24 Mart). Uzun-Kısa Vadeli Bellek(LSTM). *Medium*. 6 Nisan 2024 tarihinde <https://medium.com/deep-learning-turkiye/uzun-k%C4%B1sa-vadeli-bellek-lstm-b018c07174a3> adresinden erişildi.
- Aktivasyon Fonksiyonları Nedir? (2021, 15 Temmuz).*Devreyakan*. 4 Nisan 2024 tarihinde <https://devreyakan.com/aktivasyon-fonksiyonlari/> adresinden erişildi.
- Alabaş, M. (2019, 29 Haziran). Python İle Veri Görselleştirme: Matplotlib Kütüphanesi-1. *Medium*. 29 Mart 2024 tarihinde <https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b> adresinden erişildi.
- Asilkan, Ö. ve Irmak, S. (2009). İkinci El Otomobillerin Gelecekteki Fiyatlarının Yapay Sinir Ağları ile Tahmin Edilmesi. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14(2), 375-391.
- Asutay, A. (2023, 2 Mart). Python Kütüphaneleri: NumPy. <https://asliasutay.com/python-kutuphaneleri-numpy/> adresinden erişildi.
- Ayan, B., Kuyumcu, B. ve Ceylan, B. (2019). Twitter Üzerindeki İslamofobik Twitlerin Duygu Analizi ile Tespiti. *Gazi University Journal of Science Part C: Design and Technology*, 7(2), 495-502.
- Ayten, B. (2021, 18 Mart). Yapay Sinir Ağlarında Aktivasyon Fonksiyonları. *Medium*. 5 Nisan 2024 tarihinde <https://medium.com/databulls/yapay-sinir-a%C4%9Flar%C4%B1nda-aktivasyon-fonksiyonlar%C4%B1-11002b8ac522> adresinden erişildi.

- Balık, A. (2018, 16 Ağustos). Seaborn ile Veri Görselleştirilmesi. *Veri Bilimi Okulu*. 29 Mart 2024 tarihinde <https://www.veribilimiokulu.com/seaborn-ile-veri-gorsellestirmesi/> adresinden erişildi.
- Bilgin, M. ve Şentürk, İ. F. (2019). Danışmanlı ve Yarı Danışmanlı Öğrenme Kullanarak Doküman Vektörleri Tabanlı Tweetlerin Duygu Analizi. *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 21(2), 822-839.
- Boesch, G. (2021, 6 Ekim). VGG Very Deep Convolutional Networks (VGGNet)—What You Need to Know. *Viso.ai*. 5 Mayıs 2024 tarihinde <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/> adresinden erişildi.
- Boesch, G. (2023, 14 Kasım). Deep Residual Networks (ResNet, ResNet50)—2024 Guide. *Viso.ai*. 5 Mayıs 2024 tarihinde <https://viso.ai/deep-learning/resnet-residual-neural-network/> adresinden erişildi.
- Candan, H. (2021, 26 Aralık). Adım Adım Makine Öğrenmesi Bölüm 3: Denetimsiz Öğrenme Nedir? *Medium*. 28 Mart 2024 tarihinde <https://medium.com/machine-learning-t%C3%BCrkiye/ad%C4%B1m-ad%C4%B1m-makine-%C3%B6%C4%9Frenmesi-b%C3%B6l%C3%BCm-3-denetimsiz-%C3%B6%C4%9Frenme-nedir-f890ada49a40> adresinden erişildi.
- Cankut, A. (2023, 24 Ocak). Denetimli, Denetimsiz ve Pekiştirmeli Öğrenme. *Blog AdresGezgini*. 3 Nisan 2024 tarihinde <https://blog.adresgezgini.com/denetimli-denetimsiz-ve-pekestirmeli-ogrenme> adresinden erişildi.
- Cobanov, M. (2019, 4 Mart). Keras ile Nesne Tanıma Uygulamaları. *Medium*. 30 Mart 2024 tarihinde <https://medium.com/deep-learning-turkiye/keras-ile-nesne-tan%C4%B1ma-uygulamalar%C4%B1-5142e71aeadb> adresinden erişildi.
- Community, O. (2020, 15 Aralık). Understanding the Mechanism and Types of Recurrent Neural Networks. *Open Data Science—Your News Source for AI, Machine Learning & more*. 3 Nisan 2024 tarihinde <https://opendatascience.com/understanding-the-mechanism-and-types-of-recurring-neural-networks/> adresinden erişildi.
- Çağlar, S. (2021, 21 Eylül). Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme: Fark Nedir? *Matematiksel*. 3 Nisan 2024 tarihinde <https://www.matematiksel.org/yapay-zeka-makine-ogrenmesi-ve-derin-ogrenme-fark-nedir/> adresinden erişildi.
- Çalışkan, D. ve Demir, Ö. (2022). Derin Öğrenme Yöntemleri ile Şüpheli Davranış Tespiti. *International Periodical Of Recent Technologies In Applied Engineering*, 3(1), 28-43.
- Çetin, K.R. (2020, 4 Kasım). Yapay Sinir Ağları Google Colab Çalışma Ortamı. *Medium*. <https://ruveydakardelcetin.medium.com/yapay-sinir-a%C4%9Flar%C4%B1-google-colab-%C3%A7a%C4%B1%C5%9Fma-ortam%C4%B1-3e029af37095>
- Çolmabey, F. (2021, 26 Ağustos). Derin Öğrenme Modellerinde Aktivasyon Fonksiyonları. *Medium*. 5 Nisan 2024 tarihinde <https://medium.com/deep-learning-from-deepest/derin-%C3%B6%C4%9Frenme-modellerinde-aktivasyon-fonksiyonlar%C4%B1-124b49c96a66> adresinden erişildi.

- Data, B. (2021, 9 Şubat). Yapay Sinir Ağları ve Uygulamaları. *GTech*. 3 Nisan 2024 tarihinde <https://www.gtech.com.tr/yapay-sinir-aglari-ve-uygulamalari/> adresinden erişildi.
- Day, O. ve Khoshgoftaar, T. M. (2017). A Survey on Heterogeneous Transfer Learning. *Journal of Big Data*, 4(1), 29. doi:10.1186/s40537-017-0089-0
- Demirci, G.M., Keskin, S.R. ve Doğan. G. (2019). Sentiment Analysis in Turkish with Deep Learning. *2019 IEEE International Conference on Big Data (Big Data)*. <https://doi.org/10.1109/bigdata47090.2019.9006066>.
- Doğan, Ö. (2020, 24 Ağustos). Derin Öğrenme Nedir? Yapay Sinir Ağları Ne İşe Yarar? *Teknoloji*. 6 Nisan 2024 tarihinde <https://teknoloji.org/derin-ogrenme-nedir-yapay-sinir-aglari-ne-ise-yarar/> adresinden erişildi.
- Durna, M. B. (2019, 1 Şubat). Veri Bilimi İçin Temel Python Kütüphaneleri-1: Numpy. *Medium*. 29 Mart 2024 tarihinde <https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5> adresinden erişildi.
- El Bekri, S. (2020). *Ölçülen Kimyasal Gazların Makina Öğrenmesi ile Sınıflandırması. (Yayımlanmamış Yüksek Lisans Tezi)*. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya.
- Ergin, T. (2020, 22 Şubat). Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır? *Medium*. 3 Nisan 2024 tarihinde <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad> adresinden erişildi.
- Eyipınar, C. D., Büyükkalkan, F. ve Semiz, K. (2021). Sporcu Beslenmesi İle İlgili Youtube Video Yorumlarının Duygu Analizi. *Uluslararası Beden Eğitimi Spor ve Teknolojileri Dergisi*, 2(2), 27-39.
- Ezgi, Ç. ve İhsan, S. (2022). Using Deep Learning (CNN, RNN, LSTM, GRU) methods for the prediction of Protein Secondary Structure. *Acta Infologica*, 6(1), 43-52. doi:10.26650/acin.1008075
- Fıkırkoca, K. (2024, 14 Şubat). Transfer Öğrenme: Bilgi Mirasının Gücüyle Sıra Dışı Yolculuk. 8 Nisan 2024 tarihinde <https://ilge.com.tr/transfer-ogrenme--bilgi-mirasinin-gucuyle-sira-disi-yolculuk> adresinden erişildi.
- Göçgün, Ö. F. ve Onan, A. (2021). Amazon Ürün Değerlendirmeleri Üzerinde Derin Öğrenme/Makine Öğrenmesi Tabanlı Duygu Analizi Yapılması. *Avrupa Bilim ve Teknoloji Dergisi*, 24, 445-448.
- Hazır, E. (2021, 12 Ağustos). Supervised—Unsupervised Learning, Makine Öğrenmesi. *Medium*. 3 Nisan 2024 tarihinde <https://eneshazr.medium.com/supervised-unsupervised-learning-makine-%C3%B6%C4%9Frenmesi-b903bc09430e> adresinden erişildi.
- He, K., Zhang, X., Ren, S. ve Sun, J. (2015, 10 Aralık). Deep Residual Learning for Image Recognition. arXiv. doi:10.48550/arXiv.1512.03385

- Huang, G., Liu, Z., van der Maaten, L. ve Weinberger, K. Q. (2018, 28 Ocak). Densely Connected Convolutional Networks. arXiv. doi:10.48550/arXiv.1608.06993
- Issa, D., Demirci, M. F. ve Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, 59, 101894.
- Johnson, D. (2024, 16 Mart). Sinir Ağlarında Geriye Yayılım: Makine Öğrenimi Algoritması. 6 Nisan 2024 tarihinde <https://www.guru99.com/tr/backpropagation-neural-network.html> adresinden erişildi.
- Karabal, A. (2023). Yapay Zekada Hukuk İhlalleri. *MetaZihin: Yapay Zeka ve Zihin Felsefesi Dergisi*, 6(2), 119-147. doi:10.51404/metazihin.1269258
- Karaoğlu, A. C. (2021, 11 Haziran). Derin Öğrenme ile Derinlere Yolculuk. *Miuul Blog*. 7 Nisan 2024 tarihinde <https://miuul.com/public/not-defteri/derin-ogrenme-ile-derinlere-yolculuk> adresinden erişildi.
- Kaynar, O., Aydın, Z. ve Görmez, Y. (2017). Sentiment Analizinde Öznitelik Düşürme Yöntemlerinin Oto Kodlayıcı Derin Öğrenme Makinaları ile Karşılaştırılması. *Bilişim Teknolojileri Dergisi*, 10(3), 319-326.
- Keras ResNet50. (2023, 16 Mart). *EDUCBA*. <https://www.educba.com/keras-resnet50/> adresinden erişildi.
- Kilimci, Z. H. (2020). Borsa tahmini için Derin Topluluk Modelleri (DTM) ile finansal duygu analizi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 35(2), 635-650.
- Koçulu, B. (2021, 11 Ocak). Scikit-Learn Yapısı Kolay Açıklamalı Anlatım (Örneklerle). *Medium*. <https://burcukoculu.medium.com/scikit-learn-yap%C4%B1s%C4%B1-kolay-a%C3%A7%C4%B1klamal%C4%B1-anlat%C4%B1m-%C3%B6rneklerle-afbbb4593e5> adresinden erişildi.
- Korkusuz, R. (2019). *Futbolla İlişkin Twitter Paylaşımlarının Duygu Analiz* (Yayımlanmamış Yüksek Lisans Tezi), Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne.
- Köksal, B., Erdem, G., Türkeli, C. ve ÖZTÜRK, Z. K. (2021). Twitter’da Duygu Analizi Yöntemi Kullanılarak Bitcoin Değer Tahminlemesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9(3), 280-297.
- Microsoft Visual Studio. (2024, 28 Şubat). *Wikipedi*. https://tr.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio&oldid=31794918 adresinden erişildi.
- MobileNet-v2 convolutional neural network—MATLAB mobilenetv2. (2024). *Math Works*. 5 Mayıs 2024 tarihinde <https://www.mathworks.com/help/deeplearning/ref/mobilenetv2.html> adresinden erişildi.

- Mustafa. (2024, 8 Ocak). Python Nedir ve Nerelerde Kullanılır? *Kodcular*. <https://medium.com/kodcular/python-nedir-ve-nerelerde-kullan%C4%B1%C4%B1r-8b69afe722f9> adresinden erişildi.
- Öge, B. C. ve Kayaalp, F. (2021). Farklı Sınıflandırma Algoritmaları ve Metin Temsil Yöntemlerinin Duygu Analizinde Performans Karşılaştırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9(6), 406-416.
- Öğrekçi, S. (2020, 20 Ekim). 6. Aktivasyon Fonksiyonları. 4 Nisan 2024 tarihinde <https://sogrekci.com/ders-notu/derin-ogrenme/6-aktivasyon-fonksiyonlari/> adresinden erişildi.
- Öztürk, E. (2022, 18 Aralık). Yapay Sinir Ağları(YSA). *Patika Academy*. 3 Nisan 2024 tarihinde <https://academy.patika.dev> adresinden erişildi.
- Öztürk, Ö. F. ve Pashaei, E. (2021). Konuşmalardaki duygunun evrimsel LSTM modeli ile tespiti. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 12(4), 581-589.
- Öztürk, K. ve Şahin, M. E. (2018). Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. *Takvim-i Vekayi*, 6(2), 25-36.
- Pekel, E. (2018). *Farklı Makine Öğrenmesi Algoritmalarının Karşılaştırılması. (Yayımlanmamış Yüksek Lisans Tezi)*. Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, Samsun.
- Pirim, H. (2006). Yapay Zeka. *Yaşar Üniversitesi E-Dergisi*, 1(1), 81-93.
- Polatgil, M. (2024). TOGG Otomobilleri Youtube Video Yorumlarının Duygu Analizi ve Makine Öğrenme Modeli ile İncelenmesi. *Uluslararası Anadolu Sosyal Bilimler Dergisi*, 8(1), 15-35.
- Python Programlama Dili Nedir? Nerelerde Kullanılır? (2021, 18 Ekim). *Blue Mark Academy*. 8 Nisan 2024 tarihinde <https://bluemarkacademy.com/python-programlama-dili-nedir-nerelerde-kullanilir/> adresinden erişildi.
- Rajagopal, I. (2018, 29 Mayıs). Batch Normalization—Speed up Neural Network Training. *Medium*. 8 Nisan 2024 tarihinde <https://medium.com/@ilango100/batch-normalization-speed-up-neural-network-training-245e39a62f85> adresinden erişildi.
- Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* içinde (ss. 35-39). 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), sunulmuş bildiri, Faridabad, India: IEEE. doi:10.1109/COMITCon.2019.8862451
- Safalı, Y. (2021). *Derin Öğrenme ile Yüz Tanıma ve Duygu Analizi* (Yayımlanmamış Yüksek Lisans Tezi). Mersin Üniversitesi, Fen Bilimleri Enstitüsü, Mersin.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. ve Chen, L.-C. (2019, 21 Mart). MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv. doi:10.48550/arXiv.1801.04381
- Segal, N. (2023). Facial Expressions Training Data. *Kaggle*. 5 Mayıs 2024 tarihinde <https://www.kaggle.com/datasets/noamsegal/affectnet-training-data> adresinden erişildi.
- Sel, A. (2022). Pandemi Sürecinde Toplum Görüşünün Duygu Analizi Yöntemiyle İncelenmesi: Türkiye örneği. *Beykoz Akademi Dergisi*, 10(2), 134-154.
- Seyfioğlu, M. & Demirezen, M. (2017). A Hierarchical Approach for Sentiment Analysis and Categorization of Turkish Written Customer Relationship Management Data. In *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*. IEEE.
- Sugata, T. L. I. ve Yang, C. K. (2017). Leaf App: Leaf Recognition With Deep Convolutional Neural Networks. *IOP Conference Series: Materials Science and Engineering* içinde (C. 273, s. 012004). IOP Publishing. <https://iopscience.iop.org/article/10.1088/1757-899X/245/1/012004/meta> adresinden erişildi.
- Şahin, O. (2020, 6 Haziran). Pandas Nedir? Nasıl Kullanılır? - Python Kütüphanesi. *Teknoloji*. 29 Mart 2024 tarihinde <https://teknoloji.org/pandas-nedir-nasil-kullanilir-python-kutuphanesi/> adresinden erişildi.
- Şahinaslan, Ö., Dalyan, H. ve Şahinaslan, E. (2022). Naive Bayes Sınıflandırıcısı Kullanılarak Youtube Verileri Üzerinden Çok Dilli Duygu Analizi. *Bilişim Teknolojileri Dergisi*, 15(2), 221-229.
- Şahiner Yılmaz, Ş., Özer, İ. ve Gökçen, H. (2021). Türkçe Metinlerde Derin Öğrenme Yöntemleri Kullanılarak Duygu Analizi. *Uluslararası Bilimsel Araştırmalar ve Yenilikçi Çalışmalar Sempozyumu*, 971-982.
- Tan, M. ve Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning* içinde (ss. 6105-6114). International Conference on Machine Learning, sunulmuş bildiri, PMLR. <https://proceedings.mlr.press/v97/tan19a.html> adresinden erişildi.
- Tepecik, A. ve Demir, E. (2024). Üç Duygu ile Etiketlenmiş Türkçe Ses Kayıt Verilerinin Makine Öğrenim Algoritmalarıyla Analizi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 39(2), 709-716.
- Torres, L. F. (2023, 16 Ocak). Convolutional Neural Network From Scratch. *Medium*. 6 Nisan 2024 tarihinde <https://medium.com/latinxinai/convolutional-neural-network-from-scratch-6b1c856e1c07> adresinden erişildi.
- Turan, M. ve Arıç, E. (2021). Video Duygu Analizi. *Avrupa Bilim ve Teknoloji Dergisi, Ejosat Ek Özel Sayı (HORA)*, 34-41.

- Turan, U. N., Emre, İ. E. ve Kıran, S. (2022). Metaverse ile İlgili Türkçe Dilindeki Çeşitli Sosyal Medya Platformu Verileri ile Duygu Analizi. *Journal of Information Systems and Management Research*, 4(2), 1-16.
- Tuzcu, S. (2020). Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması. *Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi*, 1(2), 1-5.
- Türkoğlu, M. F. (2021, 20 Ağustos). Keras ve Tensorflow Nedir? *Medium*. <https://mfatih.to.medium.com/keras-ve-tensorflow-nedir-8a5b9e8d1452> adresinden erişildi.
- Uğur, B. (2020, 30 Aralık). Kivy Kullanarak Telefon Fiyatı Sınıflayıcı Geliştirme. *Medium*. 30 Mart 2024 tarihinde <https://medium.com/codeai/kivy-kullanarak-telefon-fiyat%C4%B1-s%C4%B1n%C4%B1flay%C4%B1c%C4%B1-geli%C5%9Firme-644c95443d51> adresinden erişildi.
- Uyaroğlu Akdeniz, F. N. ve Cebeci, H. İ. (2021). Belediye Hizmetlerin Değerlendirilmesinde Duygu Analizi Yaklaşımı: Sakarya İli Örneği. *Journal of Intelligent Systems: Theory and Applications*, 4(2), 127-135.
- Varol, B. (2022, 22 Ocak). Nedir Bu Yapay Zeka? #3. *Medium*. 28 Mart 2024 tarihinde <https://berikavarol.medium.com/nedi%C4%B1-bu-yapay-zeka-3-e2ff0c75712c> adresinden erişildi.
- Vural, A.G, Cambazoglu, B.B., Senkul, P., Tokgoz, Z.O. (2013). A Framework for Sentiment Analysis in Turkish: Application to Polarity Detection of Movie Reviews in Turkish. Gelenbe E., Lent R. (Edt.), *Computer and Information Sciences III*, içinde (ss. 437-445). London: Springer.
- What is EfficientNet? (2023, 17 Nisan). *Sky Engine AI*. 5 Mayıs 2024 tarihinde <https://skyengine.ai/se/skyengine-blog/121-what-is-efficientnet> adresinden erişildi.
- Yılmaz, M. C. ve Orman, Z. (2021). LSTM Derin Öğrenme Yaklaşımı ile Covid-19 Pandemi Sürecinde Twitter Verilerinden Duygu Analizi. *Acta Infologica*, 5(2), 359-372.
- Yılmaz, O. (2020, 17 Ağustos). Yapay Sinir Ağları—Aktivasyon Fonksiyonları. *Medium*. 5 Nisan 2024 tarihinde <https://medium.com/@OnurYilmaz/yapay-sinir-a%C4%B1-aktivasyon-fonksiyonlar%C4%B1-d964805b189b> adresinden erişildi.
- Yohanes, D., Putra, J. S., Filbert, K., Suryaningrum, K. M. ve Saputri, H. A. (2023). Emotion Detection in Textual Data using Deep Learning. *Procedia Computer Science*, 227, 464-473.
- Yüksel, A. S. ve Tan, F. G. (2018). Metin Madenciliği Teknikleri ile Sosyal Ağlarda Bilgi Keşfi. *Mühendislik Bilimleri ve Tasarım Dergisi*, 6(2), 324-333. <https://doi.org/10.21923/jesd.384791>
- Zhang, G. ve Ananiadou, S. (2022). Examining and Mitigating Gender Bias in Text Emotion Detection Task. *Neurocomputing*, 493, 422-434.

Zilyas, D. ve Yılmaz, A. (2023). Makine Öğrenmesi Yöntemleri ile Eğitim Başarısının Tahmini Modeli. *DÜMF Mühendislik Dergisi*, 14(3), 437-447. doi:10.24012/dumf.1322273

Zvornicanin, E. (2024, 18 Mart). What Is YOLO Algorithm? | Baeldung on Computer Science. *Baeldung*. 5 Mayıs 2024 tarihinde <https://www.baeldung.com/cs/yolo-algorithm> adresinden erişildi.

