

ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ

BİTİRME ÇALIŞMASI

**Derin Öğrenme Yöntemi Kullanarak Irk ve Dil Tanıma Tabanlı
Yolcu Bilgilendirme Sistemi**

DANIŞMAN

Doç. Dr. Gökçen ÇETİNEL

HAZIRLAYAN

Burak OKUR

İbrahim Halil ÖZÇAKIR

HAZİRAN 2024

T.C.
SAKARYA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

**DERİN ÖĞRENME YÖNTEMİ KULLANARAK IRK
VE DİL TANIMA TABANLI YOLCU
BİLGİLENDİRME SİSTEMİ**

BİTİRME ÇALIŞMASI

**Burak OKUR
B190100042
İbrahim Halil ÖZÇAKIR
B190100015**

DANIŞMAN

Doç. Dr. Gökçen ÇETİNEL

Bu çalışma .. / .. /20... tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

TEŞEKKÜR

Tez çalışmamızın her aşamasında bize rehberlik eden, bilgi ve tecrübeleriyle yol gösteren danışman hocamız Doç. Dr. Gökçen Çetinel'e şükranlarımızı sunarız. Kendilerinin değerli katkıları, bu tezin hazırlanmasında büyük bir öneme sahiptir.

İÇİNDEKİLER

TEŞEKKÜR.....	i
BEYAN.....	vi
ÖZET.....	vii
SİMGE VE KISALTMALAR LİSTESİ.....	ix
ŞEKİLLER LİSTESİ.....	x
TABLolar LİSTESİ.....	xii
BÖLÜM 1. GİRİŞ	1
1.1. Problemin Tanımı	1
1.2. Çalışmanın Amacı	2
1.3. Çalışmanın Kapsamı.....	3
1.4. Literatür Özeti	3
1.5. Çalışmanın Gerçekçi Kısıtlar Açısından Analizi	5
BÖLÜM 2. MATEMATİKSEL YÖNTEM ve TASARIM	6
2.1. Yöntem Hakkında Genel Bilgi	6
2.2. Tasarım (Yöntemin Problemin Çözümünde Kullanımı)	26
BÖLÜM 3. UYGULAMA ÇALIŞMALARI.....	52
3.1. Uygulamada Kullanılan Araç ve Gereçler	52
3.2. Uygulamanın Gerçekleştirilme Aşamaları	55
3.3. Uygulama Sonuçları ve Yorumlanması.....	60
BÖLÜM 4. SONUÇLAR VE ÖNERİLER	62
4.1. Sonuçlar	62
4.2. Öneriler.....	62
4.3. Sonuçların Sağlık, Çevre ve Güvenlik Açısından Analiz	63
BÖLÜM 5. KAYNAKLAR	64
BÖLÜM 6. ÖZGEÇMİŞ.....	69
BÖLÜM 7. EKLER.....	70
EK A. Deney Tasarımı Açıklamaları	70

EK B. IEEE Etik Kurallar Onay Formu	71
EK C. Çalışmanın Ulusal veya Uluslararası Standartlar İlişkisi	72
EK E. Çalışma Yönetimi	72

BEYAN

Bitirme alıřması raporu iindeki tm verilerin akademik kurallar erevesinde tarafımdan elde edildiđini, grsel ve yazılı tm bilgi ve sonuların akademik ve etik kurallara uygun řekilde sunulduđunu, kullanılan verilerde herhangi bir tahrifat yapılmadıđını, bařkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduđunu, alıřmada yer alan verilerin bu niversite veya bařka bir niversitede herhangi bir Tasarım/Tez alıřmasında kullanılmadıđını beyan ederim.

İmza:

İbrahim Halil ZAKIR

11/06/2024

İmza:

Burak OKUR

11/06/2024

ÖZET

İnsanların konuştuğu dili tespit eden sistemler ve insan yüzü görüntüsünden ırk tespiti yapan sistemler genellikle güvenlik sistemleri ve operatör sistemlerinde kullanılmaktadır. Ancak yapılan literatür taraması sonucunda bu iki sistemi birlikte kullanan bir çalışmaya rastlanmamıştır. Konuşulan dilin yüksek doğruluk ile tanınması için uzun (10 sn) konuşma verisi istenmektedir. Yüz görüntülerinden doğru ırk tespiti için ırk tanımlamasına uygun fazla sayıda veri setine ihtiyaç duyulmaktadır. Yapılan araştırmada yolcu terminallerinde (Havalimanı, Otogar, Tren Garı) yolcuların konforunu ve memnuniyetini arttırmak amacıyla yolcunun konuştuğu dili tespit edip bu dile uygun olarak bilgilendirme yapan yolcu bilgilendirme sistemi yapılması hedeflenmektedir. Literatür taramasında bulunan diğer çalışmalara kıyasla ırk tanıma ve dil tanıma sistemlerini bir arada kullanıp çifte doğruluk yaparak sonucun doğruluğunun arttırılması hedeflenmektedir. Çifte doğrulama yapılacağı için uzun konuşma verilerine ve fazla sayıda ırk veri setine ihtiyaç duyulmaması öngörülmüştür.

Yapılan çalışmada derin öğrenme algoritması CNN (Evrişimli Sinir Ağları) kullanılarak yüzün orta kısmındaki özelliklerin çıkarılması ve görüntü verisi Asya, Latin ve Diğer olarak üç ana ırk sınıfına ayrılması ve ses verisine göre Asya sınıfı Korece ve Çince olarak, Latin sınıfı İtalyanca ve İspanyolca olarak iki alt sınıfa ayrılması hedeflenmektedir. Ayrıca sistemde görüntüden ırk, sesten dil tanıma yapılamaması durumunda sistemin İngilizce olarak bilgi vermesi hedeflenmektedir.

Projenin ilk aşamasında dil ve görüntü veri setlerinin toplanması, ırk tanıma için uygun olmayan görsellerin çıkarılması ve görüntü boyutlarının ayarlanması gibi ön işlemler uygulanmaktadır. İkinci aşamada ön işlemlerden geçen verilerden derin öğrenme algoritması olan CNN algoritmaları yardımıyla özellik çıkarılması ve ırkların Asya ve Latin olarak sınıflandırılması yer almaktadır. Projenin üçüncü aşamasında ses veri setinde konuşulmayan kısımların atılması, konuşma verilerinin 5 sn'lik ses paketleri olarak kırılması ve spektrogramlara çevrilmesi gibi ön işlemler yer almaktadır. Dördüncü aşamada ön işleminden geçen konuşma verilerinden derin öğrenme mimarisiyle özellik çıkarma ve bu özelliklerin Korece, Çince, İspanyolca, İtalyanca

olarak sınıflandırılması yer almaktadır. Son aşamada ise bu iki sistemin birbirine entegre edilip, tasarlanan arayüz üzerinden çalıştırılması planlanmaktadır.

Projenin gerçekleştirilmesi ile yolcu terminallerinde yabancı bir ülkede yolculuk yapan insanların dil bariyerleri nedeniyle bilgilendirme alamamasının önüne geçilmiş olacak ve yolculuk deneyimlerinin olumlu yönde etkilemiş olacaktır. Bu sayede terminallerdeki personellerin iş yükü azaltılmış olacaktır.

SİMGE VE KISALTMALAR LİSTESİ

CNN	: Evrişimli Sinir Ağları
DBN	: Derin İnanç Ağları
DNN	: Derin Sinir Ağları
RBM	: Kısıtlı Boltzmann Makinesi
RNN	: Tekrarlayan Sinir Ağları
KFG	: Konfigürasyon
LR	: Öğrenme Oranı

ŞEKİLLER LİSTESİ

Şekil 1. 1.	Sistemin basit çalışma diyagramı	2
Şekil 2. 1.	Örnek CNN ağı mimarisi.....	7
Şekil 2. 2.	RNN ağı yapısı	8
Şekil 2. 3.	Oto kodlayıcı yapısı	8
Şekil 2. 4.	Kısıtlı Boltzmann makinesi yapısı.....	9
Şekil 2. 5.	Derin inanç ağı yapısı	10
Şekil 2. 6.	DNN yapısı	10
Şekil 2. 7.	Dropout uygulanmış ağ yapısı	12
Şekil 2. 8.	Batch normalizasyonu	13
Şekil 2. 9.	AlexNet katmanları.....	15
Şekil 2. 10.	GoogLeNet katmanları.....	16
Şekil 2. 11.	ResNet50 katmanları	16
Şekil 2. 12.	VGG-16 katmanları	17
Şekil 2. 13.	VGG-19 katmanları	17
Şekil 2. 14.	Konfüzyon matrisi	21
Şekil 2. 15.	Dil veri seti dağılımları.....	27
Şekil 2. 16.	Filtresiz ve filtreli ses klibi	29
Şekil 2. 17.	Spektrograma dönüştürülmüş örnek ses klibi.....	31
Şekil 2. 18.	Dil tanıma sistemi için tasarlanan CNN mimarisinin eğitim sonuç grafiği	32
Şekil 2. 19.	Dil tanıma sistemi için tasarlanan CNN katmanları	33
Şekil 2. 20.	Dil tanıma sistemi için ResNet50 mimarisinin eğitim sonuç grafiği...	34
Şekil 2. 21.	Dil tanıma sistemi KFG3 konfigürasyonunun eğitim batch'i.....	36
Şekil 2. 22.	Dil tanıma sistemi KFG3 konfigürasyonunun eğitim sonuç grafiği....	36
Şekil 2. 23.	Dil tanıma sistemi KFG3 konfigürasyonunun doğrulama seti konfüzyon matrisi sonucu	37
Şekil 2. 24.	Dil tanıma sistemi KFG3 konfigürasyonunun test seti konfüzyon matrisi sonucu	37
Şekil 2. 25.	Kırpılmış yüz verisi örneği	39
Şekil 2. 26.	İrk veri seti dağılımları	41
Şekil 2. 27.	İrk tanıma için tasarlanan CNN modelinin eğitim sonuç grafiği.....	42

Şekil 2. 28.	Irk tanıma sistemi için tasarlanan CNN katmanları.....	43
Şekil 2. 29.	Irk tanıma sistemi için GoogleNet mimarisinin eğitim sonuç grafiği .	45
Şekil 2. 30.	Irk tanıma sistemi için kullanılan VGG-16 katmanları	47
Şekil 2. 31.	Irk tanıma sistemi için VGG-16 mimarisinin eğitim sonuç grafiği	48
Şekil 2. 32.	Irk tanıma sistemi KFG4 konfigürasyonunun eğitim batch'i	50
Şekil 2. 33.	Irk tanıma sistemi KFG4 konfigürasyonunun eğitim sonuç grafiği	50
Şekil 2. 34.	Irk tanıma sistemi KFG4 konfigürasyonunun doğrulama seti konfüzyon matrisi sonucu	51
Şekil 2. 35.	Irk tanıma sistemi KFG4 konfigürasyonunun test seti konfüzyon matrisi sonucu	51
Şekil 3. 1.	PyCharm IDE arayüzü	53
Şekil 3. 2.	Tasarlanan GUI.....	57
Şekil 3. 3.	Tasarlanan GUI alan bilgisi	58
Şekil 3. 4.	Tasarlanan GUI örnek test bilgisi	59
Şekil 3. 5.	Birleştirilen sistemin çalışma algoritması.....	60
Şekil 3. 6.	Birleştirilen sistemin test sonucunda elde edilen konfüzyon matrisi....	61

TABLÖLAR LİSTESİ

Tablo 2. 1. YOLOv8-cls mimarilerinin performans karşılaştırılması [30].....	20
Tablo 2. 2. Dil veri seti içerisindeki eğitim, doğrulama ve test veri sayısı	28
Tablo 2. 3. Dil tanıma sistemi için tasarlanan CNN mimarisinde kullanılan hiper parametreler.....	32
Tablo 2. 4. Dil tanıma sistemi için ResNet50 mimarisinde kullanılan hiper parametreler.....	34
Tablo 2. 5. Dil tanıma sistemi için YOLOv8x-cls mimarisinde kullanılan farklı hiper parametreler ile eğitim sonuçları.....	35
Tablo 2. 6. FairFace veri seti toplam veri sayısı.....	38
Tablo 2. 7. Çalışmada kullanılacak ırk veri seti veri adetleri	40
Tablo 2. 8. Irk veri seti içerisindeki eğitim, doğrulama ve test veri sayısı.....	40
Tablo 2. 9. Irk tanıma sistemi için tasarlanan CNN mimarisinde kullanılan hiper parametreler.....	42
Tablo 2. 10. Irk tanıma sistemi için GoogleNet mimarisinde kullanılan hiper parametreler.....	44
Tablo 2. 11. Irk tanıma sistemi için VGG-16 mimarisinde kullanılan hiper parametreler.....	46
Tablo 2. 12. Irk tanıma sistemi için YOLOv8x-cls mimarisinde kullanılan farklı hiper parametreler ile eğitim sonuçları.....	49

BÖLÜM 1. GİRİŞ

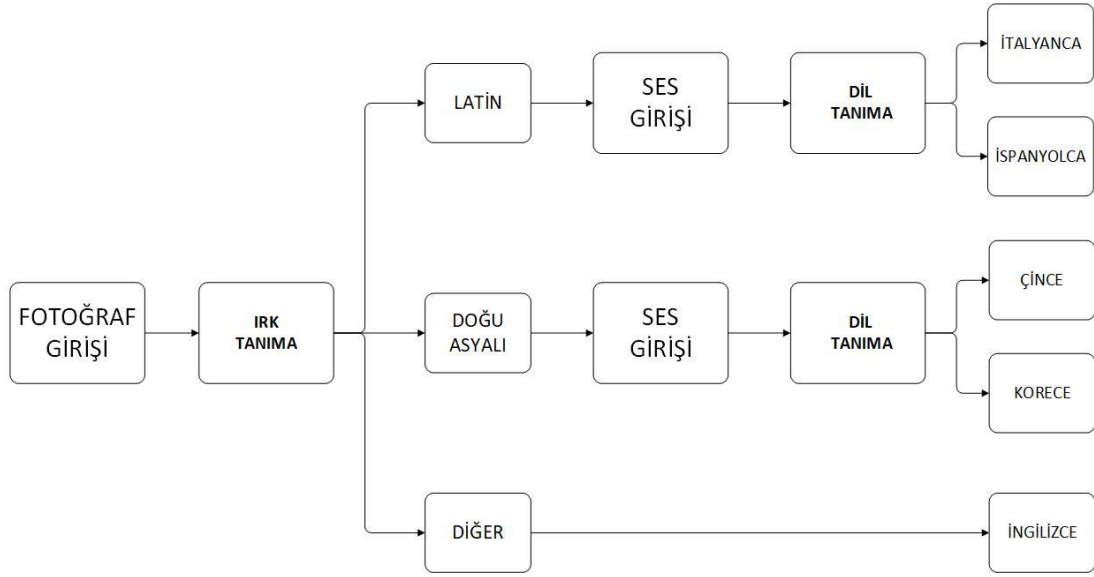
Çalışmanın giriş bölümünde, ele alınan problemin tanımlanması, amaç ve hedeflerin belirtilmesi, çalışmanın sınırlarının ve kapsamının belirlenmesi, literatürde yer alan kaynakların araştırılarak önceden yapılmış çalışmalar hakkında bilgiler sunulması hedeflenmiştir.

1.1. Problemin Tanımı

İlk zamanlarda, dil tanıma sistemleri arayan konuşmacının konuştuğu dili tespit edip bu dile uygun operatöre yönlendirecek şekilde geliştirilmiştir. Ancak, bu operatör sistemlerinde konuşulan dili tanımlamak oldukça zor olmuştur. Çünkü dil tanıma sistemlerinin yüksek doğruluk verebilmesi için 6 saniyeden fazla konuşma verisine ihtiyaç duyulmuştur. Bu sistemin yüksek doğrulukla çalışması zordur çünkü konuşmacılar genellikle telefonu ilk açtıklarında kısa kelimeler kullanmakta ve evrensel telefon açılış cümlesi olan 'Alo' kelimesi söylenmektedir. Bu kelime çoğu dilde aynı olduğu için, operatörlerde kullanılan dil tanıma sistemlerinin doğruluğu yetersiz kalmıştır.

İrk tanıma sistemleri, yüz görsellerinden bireyin hangi etnik gruba ait olduğunu tanımlama yeteneğine sahip sistemlerdir ve başlangıçta bilimsel topluluktan fazla ilgi görmemiştir. Ancak, sonraki dönemlerde yapılan çalışmalarla doğruluğun arttırılması ve yeni veri kütüphanelerinin oluşturulması sayesinde 'etnik kimlik tanımlama' araştırmalarına olan ilgi artmıştır. Bu sistemler genellikle güvenlik açısından kullanılmaktadır.

Çalışmada, ırk tanıma ve dil tanıma sistemlerinin bir arada kullanılmasıyla yüksek doğruluk elde edilmesi hedeflenmektedir. Şekil 1. 1'de görüldüğü gibi, ilk olarak alınan yüz görüntüsünden ırk sınıflandırması yapılacak, ardından alınan ses verisine göre dil sınıflandırması gerçekleştirilecektir.



Şekil 1. 1. Sistemin basit çalışma diyagramı

1.2. Çalışmanın Amacı

Çalışmada, ırk ve dil tanıma sistemlerinin bir arada kullanılması hedeflenmektedir. Tanımlanacak ırklar 3 ana ırk sınıfına indirgenmiştir. İlk olarak yüz verisine göre ırklar Asya, Latin ve Diğerleri sınıfı olarak üç sınıfa ayrılmıştır. Sonrasında ise ses verisine göre Asya sınıfındaki yolcular Korece ve Çince, Latin sınıfındaki yolcular ise İspanyolca ve İtalyanca olarak alt sınıflara ayrılmıştır. Eğer sistem yüz verisinden ırk sınıflandırması veya ses verisinden dil sınıflandırması yapamazsa, sistemin İngilizce bilgi verecek şekilde tasarlanması düşünülmüştür.

Çalışmanın amacı, daha önceden ayrı ayrı yapılan dil ve ırk tanıma sistemlerini bir arada kullanarak çifte doğrulama yapmak ve yolcunun konuştuğu dile göre yardımcı olan yolcu bilgilendirme sistemi geliştirmektir. Bu sistem, terminallerde (havalimanı, otogar, tren garı) yüksek doğrulukla çalışarak yolculara destek olmayı amaçlamaktadır.

Proje hedefleri:

1. Irk ve konuşulan dilin tanımlanması için gerekli veri setini oluşturmak.
2. Görüntü ve sesleri işleyebilmek için uygun ön işleme adımlarını gerçekleştirmek.
3. Derin öğrenme yöntemi ile verilerden özellik çıkarmak.

4. Arayüz tasarlayarak tasarlanan arayüz üzerinden bireyin görüntü ve ses girdi verisine göre konuşulan dili tespit etmek.

1.3. Çalışmanın Kapsamı

Çalışma kapsamında, önceki çalışmalarda ayrı ayrı olarak yapılan dil ve ırk tanıma sistemlerini bir arada kullanıp çifte doğrulama yaparak yüksek doğrulukla sonuç veren ve daha az veriye ihtiyaç duyan bir sistem geliştirilecektir. Çifte doğrulama sayesinde, literatürde bulunan çalışmaların aksine uzun konuşma verisine ve uygun şekilde kırılmış görüntü verisine ihtiyaç duyulmayacaktır.

Sistemde tanımlanacak ırklar, Asya, Latin ve Diğerleri olarak üç ana sınıfa ayrılacak; sonrasında ses verisine göre Asya sınıfı için Korece ve Çince, Latin sınıfı için ise İspanyolca ve İtalyanca olarak alt sınıflara ayrılacaktır.

Eğer sistem yüz verisinden ırk sınıflandırması veya ses verisinden dil sınıflandırması yapamazsa, sistem İngilizce bilgi verecek şekilde tasarlanmıştır. Tasarlanan arayüzden ses ve yüz görüntü verileri alınarak işlenecek ve Şekil 1. 1’de gösterildiği gibi sınıflandırma yapılacak ve sonuç elde edilecektir.

1.4. Literatür Özeti

Çalışmanın amacı, insanın yüz görüntüsü üzerinden görüntü işleme teknikleri ve derin öğrenme yardımıyla ırk tahmininde bulunmak ve konuşulan ses verisine göre bu tahmin sınıfları içerisinde uygun dilin tespit edilmesidir. Hem görselden hem de sestten tanıma yapılmasının amacı, doğruluk oranını artırmaktır. Literatür taraması sonucunda, görüntüden ırk tanıma ve konuşulan dilin tanımlanmasını ayrı ayrı gerçekleştiren araştırmalar bulunmuştur.

David Belcar, Petra Grd ve Igor Tomicic tarafından yapılan çalışmada, yüzün orta bölgesinden veriler alarak ırk tanımlamak için CNN (Evrişimli Sinir Ağları) kullanılmıştır. Yüzün tamamını işlemek yerine daha küçük bir kısmını alarak zamandan tasarruf edilmiştir. Ayrıca araştırmada, işlenen veriyi "landmark" kullanarak ve "landmark" olmaksızın işleyerek iki sonucun doğruluğunu

karşılaştırmışlardır. Bu araştırmada iki farklı veri kütüphanesi kullanılmış ve bu kütüphanelere göre eğitilen sistemin doğruluk oranları karşılaştırılmıştır. Modelin doğruluk oranı, beş ırk sınıflandırması için %80,34 ve yedi ırk sınıflandırması için %61,74 olarak bulunmuştur

Abdulwahid Al Abdulwahid, 2 farklı CNN modeli ve 2 farklı veri kütüphanesi kullanarak cinsiyet ve ırk tanımlaması yapmıştır. Araştırmada yüzün orta bölgesine odaklanılmıştır ve böylelikle zamandan tasarruf edilmiştir. Doğruluk yüzdelerini karşılaştırmak için Model A ve Model B olmak üzere iki farklı model tasarlanmıştır. Elde edilen sonuçlar doğrultusunda Model A'nın dört ırktaki sınıflandırma doğruluğu %85, Model B'nin ise %86 olduğu gözlemlenmiştir.

Ibrahim Hadžić, Tensorflow kütüphanesi kullanarak doğal dil tanıma araştırması yapmıştır. Araştırmanın hedefi 3 saniyelik konuşma verisinden konuşulan dili tanımayı sağlamaktır. Ses verisini spektrogramlara dönüştürerek ısı haritası olarak temsil etmiş ve bu spektrogramlardan Fourier dönüşümü alarak özellik çıkarmıştır. Bu özellikleri sınıflandırmak için Tensorflow kütüphanesi kullanılmıştır. Hırvatça, Fransızca ve İspanyolca dilleri arasında dil tanımlama yapmayı hedefleyen bu araştırma, 6, 4, 2 ve 1 saniye süren kelime öbeklerinde sırasıyla %65,3, %62.3, %52.8 ve %44.8 doğrulukla sonuç vermektedir.

Gundeep Singh, Sahil Sharma, Vijay Kumar, Manjit Kaur, MohammedBaz ve Mehedi Masud tarafından yapılan araştırmada, derin öğrenme mimarisi kullanarak konuşulan dili tanımlamayı hedeflemektedir. Kısa ses verileriyle yüksek doğruluk elde etmek için destek vektör makineleri (SVM) gibi geleneksel yöntemlerin yetersiz kaldığı belirlenmiş ve araştırma sonucunda CNN modeli önerilmiştir. Ayrıca, önceki araştırmaların ve konuşulan dil tanıma modellerinin doğruluk oranlarını da paylaşmışlardır.

Araştırma sonucunda, 22 dil üzerinde 10 saniyelik kayıt verilerinden %93 doğruluk elde edilmiştir. Literatür taraması sonucunda, dil ve ırk tanıma modellerinde daha hızlı ve yüksek doğruluk oranları elde etmek için tercih edilen yöntemlerden biri olan CNN derin öğrenme yönteminin tercih edildiği belirtilmiştir. Bu yöntem, özellik çıkarma ve sınıflandırma işlemlerinde üstün performans sağladığı için tercih edilmektedir. Ancak

arařtırmalar sonucu, konuřulan dil ve grselden ırk tanıma sistemini bir arada yapan bir arařtırma bulunamamıřtır.

nerilen alıřmada, derin ğrenme tabanlı alıřabilen ve yzn orta kısmı kullanılarak ırk tanıma ve ses verisine gre konuřulan dili tespit edebilen bir yolcu bilgilendirme sistemi yapılması hedeflenmektedir. İlk olarak, sistemin grntden ırk sınıfı tahmini yapması ve sonrasında ses girdisine gre ırk tahmin sınıfının iinden konuřulan dili tespit etmesi beklenmektedir.

1.5. alıřmanın Gereki Kısıtlar Aısından Analizi

alıřmanın gereki kısıtlar aısından analizi yapıldıėında, alıřmanın temelinde mevcut literatrde az bulunan ırk ve dil tanıma sistemlerinin arayz zerinden bir arada kullanılması yaklařımı bulunmaktadır. Bu alıřmada, bu iki sistem birleřtirilip arayz zerinden gerek zamanlı bir řekilde kullanılacaktır. Ayrıca, kullanılan veri setindeki uyumsuzluklar ve bozuk verilerin atılması sonrası veri setindeki eksik veriler sistemin performansını etkilemiřtir. Sistemin hızlı alıřması ve eėitilmesi amacıyla GPU, RAM ve Google Colab desteėi alınmıřtır. Bu sistem, yolcu terminallerinde (havalimanı, otogar, tren garı) yolcuların konforunu ve memnuniyetini arttırmak amacıyla tasarlanacak ve yabancı bir lkede yolculuk yapan insanların dil bariyerleri nedeniyle bilgilendirme alamamasının nne geecek, bylelikle yolculuk deneyimlerinin olumlu ynde artmasını saėlayacaktır.

BÖLÜM 2. MATEMATİKSEL YÖNTEM ve TASARIM

Bu bölümde çalışmada kullanılan yöntem ya da yöntemlere ait bilgilendirme ve yöntemin çalışmada kullanımına ilişkin literatür araştırmalarına yer verilmiştir. Yöntem veya yöntemlerin problemin çözümünde kullanımı ve tasarım aşamaları da bu bölümde verilmektedir.

2.1. Yöntem Hakkında Genel Bilgi

Derin öğrenme, yapay sinir ağları oluşturarak verileri işleyen ve bu verilerden tahminler üreten bir yapay zekâ tekniğidir. Literatürde ve mevcut araştırmalarda derin öğrenmenin geniş bir uygulama alanı olduğu görülmüştür. Doğal dil işleme, görüntü işleme, ses işleme, nesne tanıma, robotik ve otonom araç sistemleri gibi alanlar, derin öğrenme yöntemlerine dayanmaktadır. Çalışmada ele alınan ırk ve dil tanıma sistemleri de derin öğrenmeye dayanmaktadır.

- **Derin Öğrenme Modelleri**

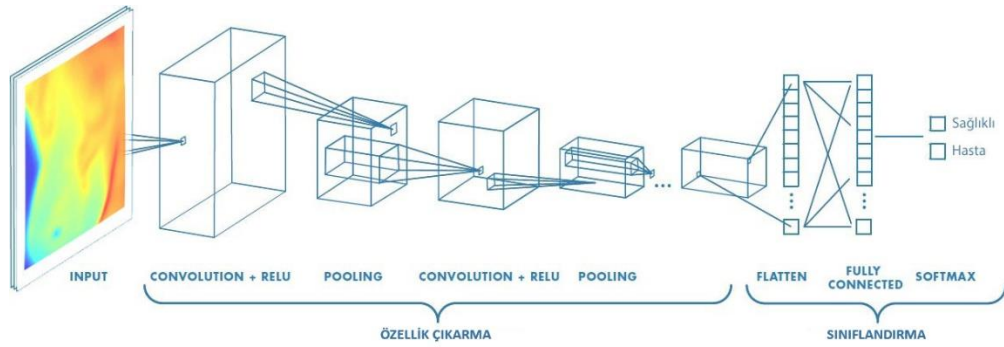
Derin öğrenme terimi, genelde veri analizinde kullanılan bir yöntemdir. Bu yöntem, verinin özelliklerini temsil ederek öğrenmeye dayanmaktadır. Derin öğrenmenin birden fazla modeli vardır ve her bir model, farklı problemlere uygun olabilmektedir. Örneğin, sınıflandırma, tespit, tahmin ve teşhis gibi farklı sonuçlar elde etmek için farklı yapay sinir ağı modelleri tercih edilebilmektedir. Ayrıca, hangi modelin kullanılacağı verinin türüne bağlı olarak değişebilmektedir. Örneğin, görüntü, ses veya sinyal verisi gibi farklı veri türleri, uygun sinir ağı modelinin seçiminde etkili olabilmektedir. Derin öğrenme çalışmalarında genellikle ağ mimarisi, hiper parametre seçimi ve optimizasyon stratejisi gibi ana konular üzerinde durulmaktadır. Bu çalışma kapsamında, 6 farklı derin öğrenme modeli incelenmiştir.

- **Evrişimsel Sinir Ağları (CNN)**

Evrişimsel sinir ağları (CNN), özellikle iki boyutlu görüntü verileri üzerinde çalışmak için tasarlanmış derin öğrenme yöntemidir. Diğer sinir ağlarına kıyasla daha az sayıda nöron bağlantısı içermektedir. CNN'ler, görüntü analizlerinde ve sınıflandırma işlemlerinde yüksek başarı gösterirken, doğal dil işleme alanında da etkili sonuçlar

elde etmiştir. Bu ağlar, hayvan görme sistemine dayanan bir yaklaşımla Hubel ve Wiesel tarafından öne sürülmüş ve LeCun ve ekibi tarafından gradyan temelli bir yöntemle geliştirilmiş, ilk CNN mimarisi olan LeNet'i oluşturmuştur [10]. Filtreleme esasına dayanan bir yapıya sahip olan evrişimsel sinir ağları, görüntülerin özniteliklerini belirginleştirmek için çeşitli filtreler kullanmıştır. Bu filtreler, farklı boyut ve değerlere sahip olabilir ve görüntüdeki belirsiz öznitelikleri ortaya çıkararak sınıflandırma işlemlerinde başarı sağlamaktadır. Örnek CNN ağı mimarisi Şekil 2.1'de verilmiştir.

Çok katmanlı sinir ağı, aktivasyon katmanı, birden fazla konvolüsyon katmanı, sınıflandırıcı katman, tam bağlı katman, havuzlama katmanı ve ek katmanlar gibi bileşenlerden oluşmaktadır. Her bir katman, özel bir işlevi yerine getirerek sınıflandırıcı katmanda sonuç üretmektedir. Evrişimsel sinir ağları, derin öğrenme yapıları içinde en yaygın kullanılan sinir ağlarıdır ve genellikle sınıflandırma ve tespit işlemleri için tercih edilebilmektedirler. Bu ağlardaki katmanlar, sınıflandırılacak öğelerin özniteliklerini belirleyerek bu öğeleri sınıflandırıcı katmanda sınıflandırmaktadırlar.

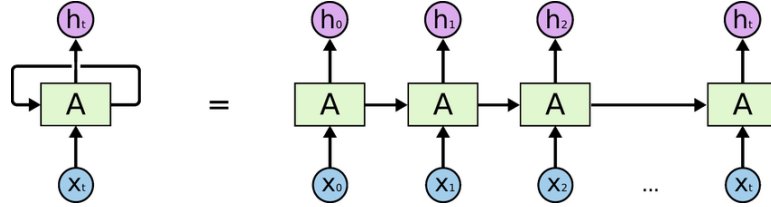


Şekil 2. 1. Örnek CNN ağı mimarisi

- **Tekrarlayan Sinir Ağları (RNN)**

Tekrarlayan sinir ağları, sıralı veriler için tasarlanmış ağlardır ve genellikle doğal dil işleme alanında kullanılmaktadırlar. RNN, yalnızca ağa giren giriş örneklerini değil, aynı zamanda önceki zaman serisi içindeki giriş örneklerini de kullanmaktadır. RNN ağının amacı, ardışık olarak gelen verileri kullanmaktır. Diğer sinir ağlarının girişleri birbirinden bağımsız bir şekilde ağa girerken, RNN ağlarında dizinin her çıktısı önceki hesaplamalara bağlıdır. RNN ağı, doğal dil işlemede yaygın olarak kullanıldığı gibi,

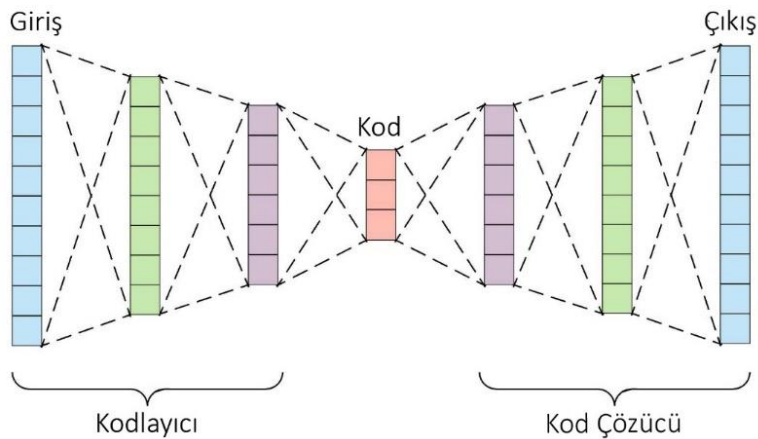
ardışık bir dizi yapısından sonraki noktayı tahmin etme işleminde de kullanılmaktadır. Örneğin, cümlede boş bırakılan kısmı tahmin etmek için kullanılabilir [9]. RNN ağı yapısı Şekil 2. 2’de verilmiştir.



Şekil 2. 2. RNN ağı yapısı

• Derin Oto Kodlayıcılar

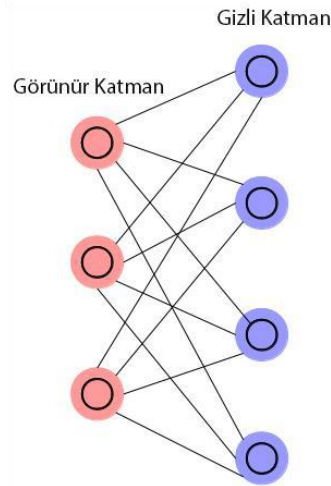
Oto kodlayıcılar, veriden otomatik olarak gerekli öznetelik kümesini çıkarmak amacıyla geliştirilmiş sinir ağlarıdır. Bir oto kodlayıcı, girdi vektörüne sınıf etiketi vermez, bunun yerine onu yeniden oluşturmaktadır. Derin öğrenmenin bir parçası olan derin oto kodlayıcılar ise, yüksek boyutlu verileri temsil etmek için birden fazla oto kodlayıcının üst üste eklenmesiyle oluşturulan yapılar kullanmaktadır [11]. Derin oto-kodlayıcıların temel amacı, özellik kümesini otomatik olarak çıkarmak veya veri boyutunu azaltmaktır. Kodlayıcı, ham verilerden özellikleri çıkarır ve kod çözücü, çıkarılan özelliklerden verileri yeniden oluşturmaktadır. Eğitim sırasında, kodlayıcının girdisi ile kod çözücünün çıktısı arasındaki farklılık kademeli olarak azaltılmaktadır. Bu yöntem, denetimsiz bir öğrenme yaklaşımı olduğu için, etiketli veri kümesine ihtiyaç duymamaktadır. Ancak, yöntemin doğru ağırlıkları belirleyebilmesi için bir ön öğrenme aşaması gerekmektedir. Ön öğrenme aşamasında, uygun ağırlıklar yaklaşık olarak yönteme girilmektedir. Oto kodlayıcı yapısı Şekil 2. 3’te verilmiştir.



Şekil 2. 3. Oto kodlayıcı yapısı

- **Kısıtlı Boltzmann Makinesi (RBM)**

Boltzmann makineleri, Boltzmann dağılımına uyum sağlayan rastgele konumlandırılmış bir sinir ağıdır. Sınıflandırma, özellik öğrenimi ve regresyon işlemlerini yapabilen Boltzmann makineleri, giriş veri seti üzerindeki olasılıksal dağılımları öğrenebilen sinir ağlarıdır. Bir görünür katman ve bir gizli katmandan oluşmaktadır. Aynı katmandaki birimler bağlı değildir, ancak farklı katmanlardaki birimler birbirine bağlıdır. Ağ katmanları arasında yönsüz bir bağlantı vardır, bu nedenle her iki yöndeki ağırlıklar aynıdır. Görünür katmanda girdiler hesaplanmakta ve bu girdilerin bir sonraki düğüme iletilip ileilmeyeceği rastgele olarak belirlenmektedir. Derin Boltzmann makineleri, art arda eklenmiş kısıtlı Boltzmann makineleri katmanlarından oluşan bir sinir ağıdır. Zaman karmaşıklığının fazla olmasından dolayı büyük veri kümelerinde öğrenme süreci yavaştır [12]. Kısıtlı Boltzmann makinesi yapısı Şekil 2. 4'te verilmiştir.

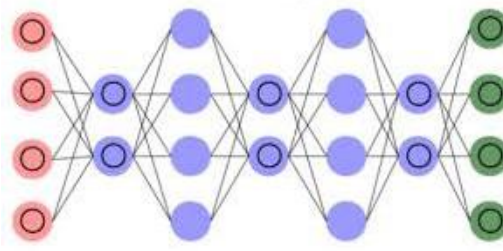


Şekil 2. 4. Kısıtlı Boltzmann makinesi yapısı

- **Derin İnanç Ağları (DBN)**

DBN, art arda sıralanmış kısıtlı Boltzmann makinelerinin katmanlarının birleştirilmesi ile oluşan sinir ağıdır. RBM algoritması ise rastgele bir sinir ağı olarak tanımlanmaktadır. Gaussian dağılımına sahip rastgele birimler içermektedir. DBN, Kısıtlı Boltzmann makinelerinin katmanlarının sırasıyla eğitilerek öğrenilmesiyle gerçekleşir. Her RBM'nin gizli katmanı, bir sonraki RBM'nin görünür katmanına bağlanmıştır. Yoğun seviyede yönsüz bağlantı bulunmaktadır. DBN, denetimli ve denetimsiz öğrenme amacıyla kullanılabilir. Ağın eğitime başlamak için açgözlü bir algoritma gerekmektedir. Açgözlü algoritma, her adımda mevcut en iyi

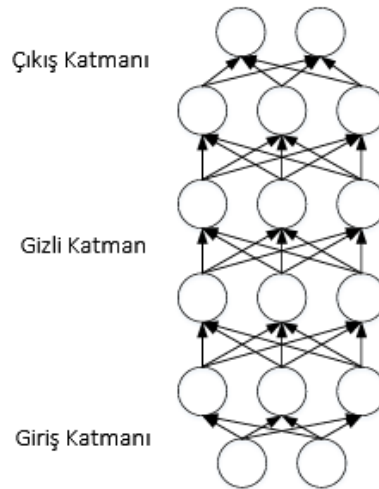
seçimi yaparak en iyi çözümü bulmayı amaçlayan algoritmadır [13]. Derin inanç ağı yapısı Şekil 2. 5'te verilmiştir.



Şekil 2. 5. Derin inanç ağı yapısı

- **Derin Sinir Ağları (DNN)**

Sinir ağlarının geliştirilmeye başlandığı ilk aşamada bir girdi katmanı mevcuttur ve direkt olarak çıkışa bağlıdır. Bu derin öğrenme algoritmasına birden fazla gizli katman eklenerek daha karmaşık problemler çözülmeye çalışılmış ve her katmanın ağırlığı, delta kuralı adı verilen bir öğrenme yöntemiyle ayarlanmıştır. Bu tür sinir ağlarına daha fazla gizli katman eklenmesiyle, karmaşık ve doğrusal olmayan ilişkilerin de tespit edilmesi sağlanmıştır. Bu sinir ağlarına "derin sinir ağları" (DNN) adı verilmektedir. DNN'ler, hem denetimli hem de denetimsiz öğrenme problemleri için kullanılabilir. Denetimsiz öğrenme aşamasında bir DNN eğitilirken parametreler ilk olarak etiketsiz veriler kullanılarak eğitilmektedir. Denetimli öğrenme aşamasında ise sinir ağına etiketli verilerle ince ayar yapılmaktadır. [14]. DNN yapısı Şekil 2. 6'da verilmiştir.



Şekil 2. 6. DNN yapısı

Derin Öğrenme Katmanları

- **Konvolüsyon Katmanı**

Evrişimsel sinir ağlarının temel bileşeni olan bu katman, girdi verisi üzerinde belirli filtrelerin uygulanmasıyla çalışmaktadır. Veri bir konvolüsyon katmanına girdiğinde katman tüm filtreleri girdinin boyutları boyunca katlamakta ve 2D bir aktivasyon haritası oluşturmaktadır [15]. Filtreler, girdinin belirli özelliklerini ortaya çıkarmak amacıyla verinin üzerinde gezdirilmektedir. Filtreleme işlemi sonucunda, giriş verisinden daha küçük boyutlarda bir matris elde edilmektedir. Uygulanacak filtreler $M \times M$ türünde olmalıdır. Tercih edilen filtre boyutu, ağın başarısı ve eğitim süresi üzerinde değişikliklere neden olabilmektedir.

- **Aktivasyon (ReLU) Katmanı**

Verilerdeki karmaşık ve doğrusal olmayan ilişkileri tespit etmek ve öğrenmek için doğrusal olmayan dönüşümlere ihtiyaç duyulmaktadır. Aktivasyon fonksiyonları, ağın doğrusal olmayan ilişkilerini yakalamasını sağlamaktadır. Bu fonksiyonlar, verileri belirli bir aralığa sıkıştırmaktadır. Sigmoid, hiperbolik tanjant, softmax ve ReLu gibi çeşitli aktivasyon fonksiyonları bulunmaktadır, ancak en yaygın olarak kullanılan ReLu fonksiyonudur. ReLu fonksiyonu, sonuç olarak elde edilen değerin negatif olması durumunda 0, pozitif olması durumunda ise değerin kendisini almaktadır [16].

$$f(x) = \max(0, x), \text{ ReLu Fonksiyonu} \quad (2.1)$$

- **Havuzlama Katmanı**

Havuzlama katmanı, ağın içindeki bir sonraki katman için giriş boyutunu azaltarak verileri indirgemektedir. Bireysel özellik kanalları üzerinde çalışarak yerel bir bölgedeki veriyi toplamakta ve tek bir değere dönüştürmektedir. Havuzlama işlemi sonrasında verilerde kayıp oluşmaktadır, ancak veri miktarındaki azalma eğitimin daha hızlı olmasını sağlamaktadır. $M \times M$ boyutlu bir filtre, veri havuzunda dolaşarak matriste en uygun değeri bulmaktadır. Yaygın olarak kullanılan yöntemler arasında maksimum operatörünü kullanan "maksimum havuzlama" ve ortalama operatörünü kullanan "ortalama havuzlama" bulunmaktadır [17]. Maksimum havuzlamada, $M \times M$ boyutlu matriste bulunan en büyük değer, oluşturulacak yeni veri matrisinin 1×1 boyutundaki değerini oluşturmaktadır. Ortalama havuzlama işleminde ise $M \times M$

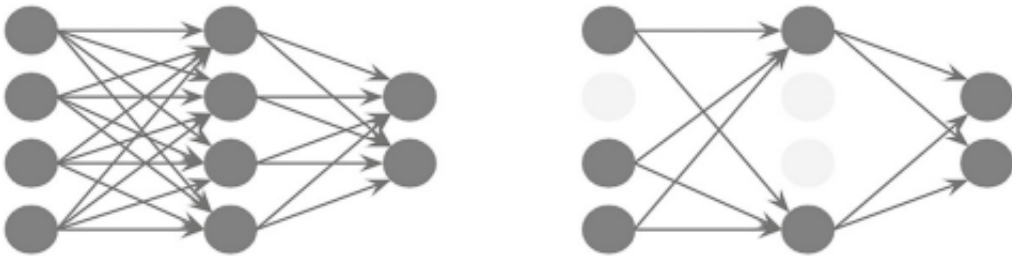
boyutundaki matrisin tüm değerlerin ortalaması alınarak yeni veri matrisinin 1×1 boyutundaki değeri oluşmaktadır.

- **Tam Bağlı Katman**

Bu katmandaki tüm nöronlar dizi şeklinde görünmekte ve önceki katmandaki tüm aktivasyonlara tam bağlı olmaktadır. Tam bağlı katmanın özelliği, önceki katmanların çıkışına bağlı olarak ortaya çıkmaktadır. Veriyi belirleyecek olan özelliklerin hangi sınıfla ilişkili olduğu belirlenmektedir. Bir tam bağlı katman, yüksek seviyeli özelliklerin hangi sınıfa ait olduğunu belirlemek için bu özellikleri temsil eden ağırlıklara bakmaktadır. [12].

- **Dropout Katmanı**

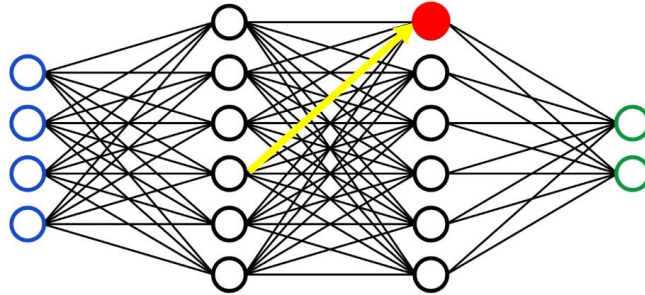
Aşırı uyum, derin öğrenme modelinin eğitim verisine çok iyi uyum sağlaması ancak test verilerinde iyi performans göstermemesidir. Aşırı uyum gösteren bir modelde, eğitim verisinde yüksek doğruluk oranı elde edilirken, test verisinde veya gerçek dünyadaki verilerde düşük doğruluk oranları elde edilmektedir. Dropout bu aşırı uyum durumunu azaltmak için kullanılmaktadır [18]. Ağın verileri ezberlemesi durumunu ortadan kaldırmak için bazı ağ düğümlerini ortadan kaldırmak gerekmektedir. Dropout katmanı tam bağlı katmanları düzenlemek için kullanılmaktadır. Her eğitim katmanı için birimleri rastgele kaldırarak modeli tekrar düzenlemektedir. Dropout uygulanmış ağ yapısı Şekil 2. 7'de gösterilmiştir. Dropout 'un önerilen değeri iç katmanlar için genellikle 0.5 ila 0.8 arasındadır.



Şekil 2. 7. Dropout uygulanmış ağ yapısı

- **Batch Normalizasyon Katmanı**

Bir sinir ağı eğitimi sırasında her katmanın giriş değerlerinin dağılımı, ondan önce gelen tüm katmanlar tarafından etkilenmektedir. Bu değişkenlik, eğitim hızını azaltmaktadır. Batch normalizasyonu, bu değişkenliği çözmek ve öğrenmeyi hızlandırmak için geliştirilmiştir. Her bir verinin ağı giriş katmanında kullanılmadan önce değerinin normalize edilmesi bilinen bir tekniktir. Batch normalizasyon bir adım daha ileri giderek yalnızca giriş katmanını değil, ağı her katmanını normalize etmektedir. Batch normalizasyonu, her mini-Batch için birim değerlerini ayarlayarak çalışmaktadır. Mini-batch'ler eğitim sırasında rastgele oluşturulduğu için, her seferinde farklı mini-batch'ler üzerinde normalizasyon yapılmaktadır. Bu rastgelelik, eğitim sürecinde belirli bir düzeyde gürültü oluşmasına neden olmaktadır. Bu gürültü, modelin aşırı uyum yapmasını engelleyerek düzenleyici bir etki göstermektedir. Batch normalizasyonunun bu düzenleyici etkisi, dropout tarafından sağlanan etkiye benzemektedir. Bu etki dropout kullanımını gereksiz hale getirebilmektedir [18]. Batch normalizasyonu Şekil 2. 8'de verilmiştir.



Şekil 2. 8. Batch normalizasyonu

- **Sınıflandırma (Softmax) Katmanı**

Sınıflandırma katmanı olarak kullanılmaktadır. Kendisinden önceki bağlı katmanların çıkış verisini girdi verisi olarak kullanarak sınıflandırma yapmaktadır. Bunun başlıca nedeni matematiksel modelinin, verilen bir girdinin mevcut kategorilerden her birine ait olma olasılıklarını çıkarıp sınıflandırma görevine oldukça uygun olmasıdır. Bu fonksiyon, olası kategori sayısına eşit sayıda nörondan oluşmaktadır ve her nöronun çıktısı belirli bir kategorinin olasılığıdır [19]. Yumuşatma fonksiyonu ağ tarafından üretilen gerçek değerleri olasılıklara dönüştürmek için katmandaki her nörona uygulanmaktadır.

Hiper Parametreler

Derin öğrenme modellerinin eğitim sürecinde kullanılan optimizier, öğrenme oranı, epok, batch boyutu adı verilen parametrelere hiper parametreler denir. Hiper parametreler, modelin eğitim aşamasında ayarlanan ve modelin performansını önemli ölçüde etkileyen parametrelerdir. Hiper parametrelere değerlerinin değiştirilmesi modelin başarısını ve genelleme yeteneğini doğrudan etki etmektedir [27].

- **Epok**

Epok, modelin eğitim için kullanılan veri seti üzerinden bir tam geçiş yapması anlamına gelmektedir. Eğitim sürecinde model, veri seti üzerinde birden fazla epok sayısınca eğitilmektedir. Epok sayısı, genellikle modelin yeteri kadar iyi öğrenmesi ve veriler üzerinde genelleme yapabilmesi için yeterli olmalıdır, ancak çok fazla epok sayısı aşırı uyuma yol açabilmektedir.

- **Öğrenme Oranı**

Öğrenme oranı, her adımda ağırlıkların ne kadar güncelleneceğini belirlemektedir. Eğitim modeli ağırlıkları minimum kayıp fonksiyonuna doğru ilerlerken her yinelemede adım boyutunu belirleyen ayar parametresidir. Çok yüksek bir öğrenme oranı modelin hızlı bir şekilde eğitim verilerine uyum sağlamasına ve aşırı uyuma neden olabilmektedir. Çok düşük bir öğrenme oranı ise modelin çok yavaş öğrenmesine veya lokal minimumda sıkışmasına neden olabilmektedir.

- **Batch Boyutu**

Batch boyutu, modelin ağırlıklarını güncellemek için her seferinde kaç eğitim örneğinin kullanılacağını belirlemektedir. Batch boyut değeri, veri setindeki verilerin eğitime kaç adet veri ile gireceğini belirlemektedir. Her adımda batch boyut değeri kadar veri eğitime girmektedir. Küçük batch boyutları daha gürültülü gradiyent tahminlerine neden olabilmekte, ancak daha hızlı öğrenme sağlamaktadır. Büyük batch boyutları ise daha doğru gradiyent tahminleri yapmakta ancak bu durum daha fazla hesaplama gerektirmekte ve eğitim süreci daha yavaş olabilmektedir.

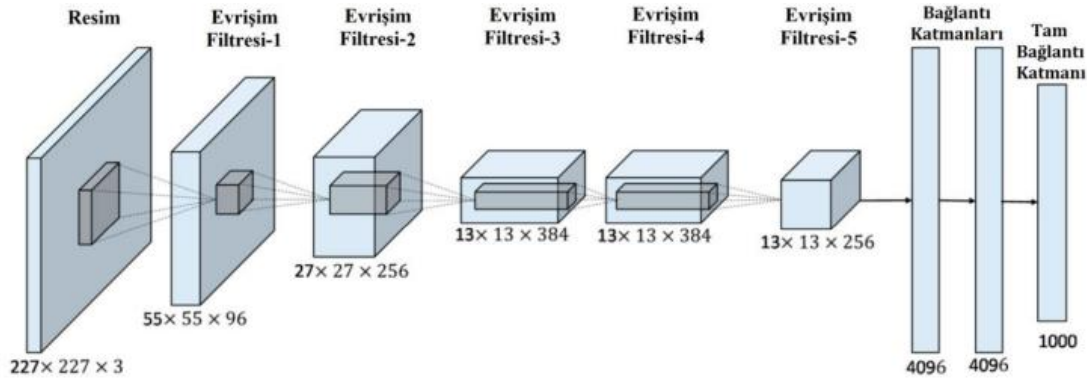
- **Optimizer**

Optimizasyon algoritmaları, modelin ağırlıklarını güncelleyerek kayıp fonksiyonunu minimize etmeye çalışmıştır. Derin öğrenme çalışmalarında yaygın olarak SGD, Adam, RMSprop ve Adamax algoritmaları kullanılmaktadır.

Derin Öğrenme Mimarileri

- **AlexNet**

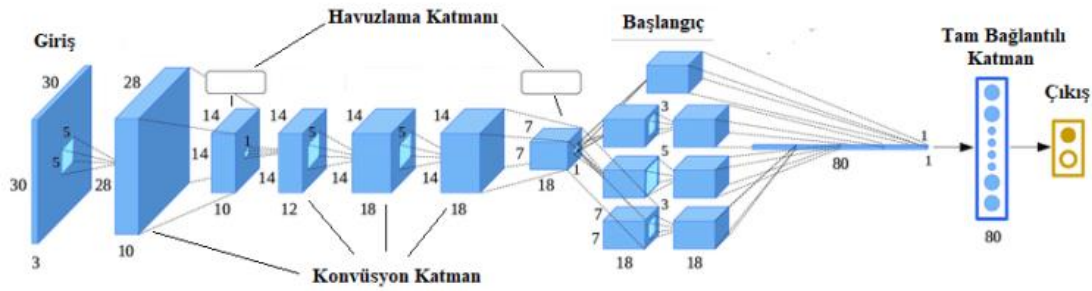
AlexNet, 8 katman içeren bir CNN modelidir. AlexNet mimarisi içerisinde Şekil 2. 9’da verilmiş olan beş konvolüsyon katmanı ve üç tam bağlantılı katman bulundurmaktadır. İlk konvolüsyon katmanında, 11×11 boyutunda 96 farklı filtre kullanılarak konvolüsyon ve maksimum havuzlama ile batch normalizasyonu gerçekleştirilmiştir. Giriş, boyutu 227×227 olan bir RGB görüntü olmalıdır. Bu görüntü boyutu olmadan, AlexNet önemli ölçüde aşırı uyum göstermiştir [20].



Şekil 2. 9. AlexNet katmanları

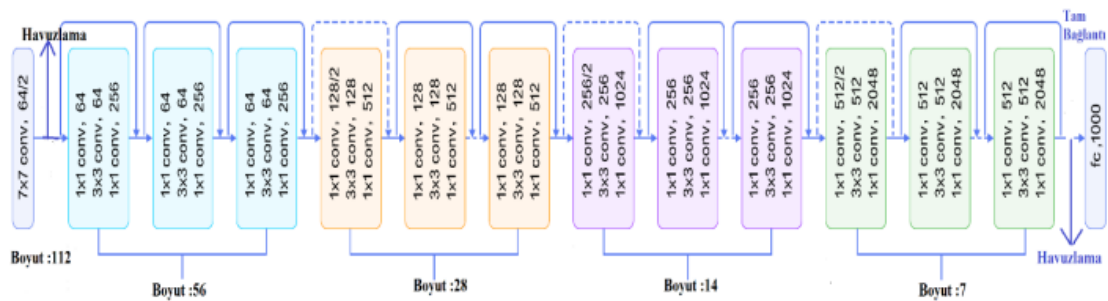
- **GoogleNet**

Bir Evrişimsel Sinir Ağı'nın performansını artırmanın en iyi yolu, daha fazla katman eklemek ve katmanlara daha fazla parametre eklemektir. Ancak, bu iki büyük soruna yol açmaktadır. Birincisi, çok fazla parametre aşırı uyuma neden olmakta ikincisi, modelin eğitim zamanı artmaktadır. GoogleNet mimarisi karmaşık bir yapısı vardır. GoogleNet'in ana derinliği 22 katmandır, ancak tüm paralel yapıların ve ek katmanlar ile toplamda 144 katman içermektedir. Diğer derin öğrenme mimarilerinde bulunan ardışık katman yapısından farklı olarak, GoogleNet derinlikli bir yapı sunmaktadır [12]. GoogleNet katmanları Şekil 2. 10’da verilmiştir.



- **ResNet50**

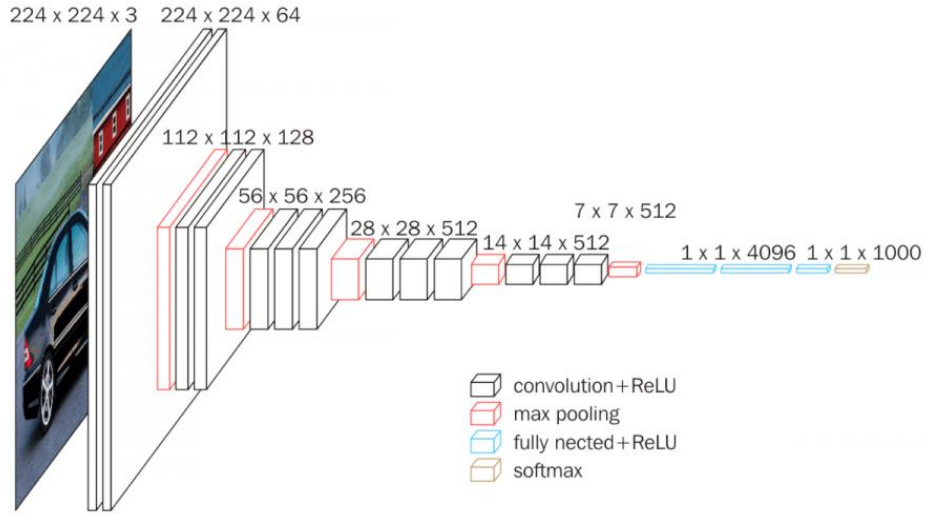
ResNet50, çeşitli görüntü işleme görevlerinde öncelikle kullanılan ve derinliği 50 katmandan oluşan bir evrimsel sinir ağı mimarisidir. ResNet50 mimarisinde kırk sekiz konvolüsyon, bir maksimum havuzlama ve bir ortalama havuzlama katmanı bulunmaktadır [21]. Bazı katmanlar arasındaki değişiklikler göz ardı edilerek alt katmana geçiş yapması ResNet50'yi ön plana çıkarmaktadır. ResNet50 mimarisinde bu durum izin verilerek başarı oranı daha üst seviyelere çıkarılmıştır. Giriş katmanı $224 \times 224 \times 3$ boyutundadır. ResNet50 katmanları Şekil 2. 11'de verilmiştir.



- **VGG-16**

VGG-16, 16 katmanlı bir CNN mimarisidir. Bu mimari çok fazla sayıda parametreye sahip değildir. Diğer gelişmiş mimarilere göre daha az parametre içermektedir. VVG16 model mimarisi, 3×3 çekirdek boyutuna sahip CNN katmanlarına odaklanmaktadır. Mimarinin içerisinde maksimum havuzlama katmanı, tam bağlı katman, ReLu katmanı, dropout katmanı ve softmax katmanı yer almaktadır [22]. Son katman sınıflandırma katmanıdır. Bu mimarinin minimum beklenen giriş görüntü boyutu, 224×224 pikseldir. Bu mimarinin önemli yanı katmanlarının erişilebilir olması

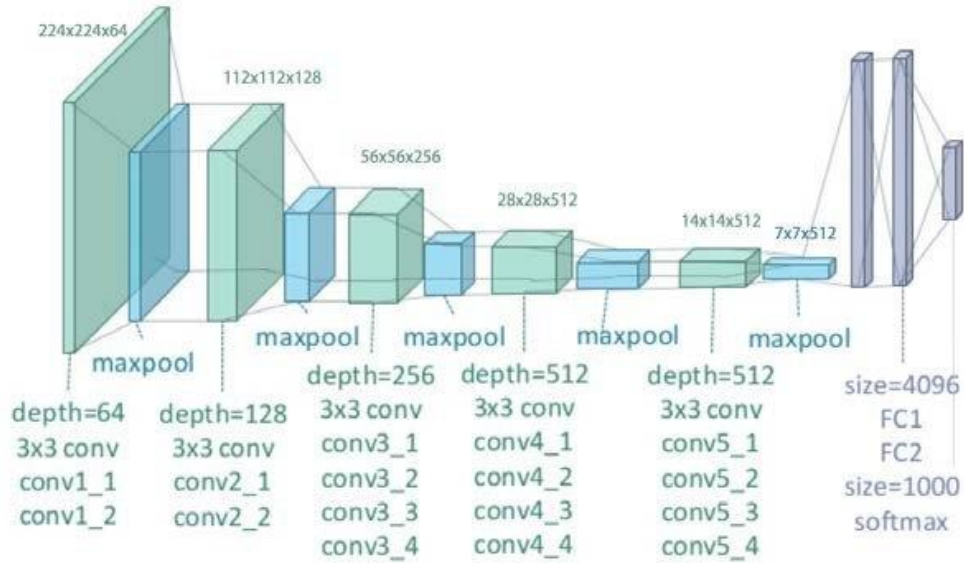
ve sistemlerde ve uygulamalarda kullanılmak üzere indirilebilmesidir. VGG-16 katmanları Şekil 2. 12’de verilmiştir.



Şekil 2. 12. VGG-16 katmanları

• VGG-19

VGG-19 mimarisi, evrişimsel katman, maksimum havuzlama, tam bağlı katmanlar ve dropout dahil olmak üzere 19 eğitilebilir katman içermektedir [23]. VGG-19, VGG-16'nın bazı değişikliklerle genişletilmiş bir versiyonudur. VGG-16 mimarisindeki 16 katmana kıyasla 19 katmana sahiptir. Bu mimarinin VGG-16 mimarisinde olduğu gibi beklenen giriş görüntü boyutu, 224×224 pikseldir. VGG-19 katmanları Şekil 2. 13'te verilmiştir.



Şekil 2. 13. VGG-19 katmanları

- **YOLOv8-cls**

YOLOv8-cls, Ultralytics tarafından geliştirilen son teknoloji ürünü bir nesne algılama algoritmasıdır. YOLOv5 algoritmasını temel alır, ancak bir dizi iyileştirme ile güncellenmiştir [28]. Yapılan güncellemelerden bazıları verilmiştir.

- Daha verimli ve doğru olan yeni bir omurga mimarisi.
- Nesne tespitinin doğruluğunu artırmaya yardımcı olan yeni bir boyun mimarisi.
- Daha doğru nesne sınıflandırması sağlayan yeni bir baş mimarisi

Desteklenen Görevler ve Farklı Varyantlar

YOLOv8-cls serisi, her biri bilgisayarla görme alanındaki belirli görevler için uzmanlaşmış çeşitli mimariler sunmaktadır. Bu mimariler, nesne algılamadan örnek segmentasyonu, poz/anahtar nokta algılama, yönlendirilmiş nesne algılama ve sınıflandırma gibi daha karmaşık görevlere kadar çeşitli gereksinimleri karşılamak üzere tasarlanmıştır.

YOLOv8-cls, nesne algılama alanında önemli gelişmeler sunan çeşitli model varyantları içermektedir. Bu varyantlar, belirli görevler ve veri setleri için optimize edilmiştir ve her biri farklı performans ve doğruluk seviyeleri sunmaktadır. YOLOv8n (nano), serinin en hafif modelidir ve düşük hesaplama gücüne sahip cihazlar için optimize edilmiştir, hızlı ve verimli çalışmaktadır. YOLOv8s (small), daha fazla hesaplama gücü gerektirmektedir ancak daha yüksek doğruluk sağlamaktadır. YOLOv8m (medium), orta seviye bir model olup dengeleyici bir performans ve doğruluk sunmaktadır. YOLOv8l (large) ve YOLOv8x (extra-large), serinin en büyük ve en karmaşık modelleridir; en yüksek doğruluk oranlarını sağlamaktalar ancak daha fazla hesaplama kaynağı gerektirmektedirler. Tablo 2. 1’de bu modellerin ImageNet veri seti üzerindeki performans metrikleri verilmiştir. Bu karşılaştırma, araştırmacılara ve uygulayıcılara, çalışmada kullanacakları en uygun modeli seçme konusunda rehberlik edecektir [30].

YOLOv8-cls hiper parametreleri:

- **hsv_h**: Renk çarkının bir kesir oranıyla görüntünün renk tonunu ayarlamakta, farklı aydınlatma koşullarında modelin genelleştirme yapmasına yardımcı olmaktadır.
- **hsv_s**: Görüntünün doygunluğunu bir kesir oranıyla değiştirmekte, renklerin yoğunluğunu etkilemektedir. Farklı çevresel koşulları simüle etmek için faydalıdır.
- **hsv_v**: Görüntünün değerini (parlaklık) bir kesir oranıyla değiştirmekte, modelin farklı aydınlatma koşullarında iyi performans göstermesine yardımcı olmaktadır.
- **optimizer**: Eğitim için optimizier seçimi. SGD, Adam, AdamW, NAdam, RAdam, RMSProp vb. gibi seçenekler veya model yapılandırmasına göre otomatik seçim için 'auto'. Yakınsama hızını ve istikrarını etkilemektedir.
- **Lr0**: Başlangıç öğrenme oranı (örneğin, SGD=0.01, Adam=0.001). Bu değerin ayarlanması, model ağırlıklarının ne kadar hızlı güncelleneceğini etkilemekte ve optimizasyon süreci için kritiktir.
- **imgsz**: Eğitim için hedef görüntü boyutu. Tüm görüntüler, modele beslenmeden önce bu boyuta yeniden boyutlandırılmaktadır. Modelin doğruluğunu ve hesaplama karmaşıklığını etkilemektedir.
- **batch**: Eğitim için toplu boyut, modelin iç parametreleri güncellenmeden önce kaç görüntünün işlendiğini belirtmektedir.
- **epochs**: Toplam eğitim epoklarının sayısı. Her epok, tüm veri kümesinin üzerinden tam bir geçiş temsil etmektedir. Bu değeri ayarlamak, eğitim süresini ve model performansını etkileyebilmektedir.

- scale: Görüntüyü bir ölçekleme faktörüyle yeniden boyutlandırmakta, kameradan farklı mesafelerdeki nesneleri simüle etmektedir [34].

Tablo 2. 1. YOLOv8-cls mimarilerinin performans karşılaştırılması [30]

Model	Boyut(piksel)	doğruluk top1	doğruluk top5	GPU Hızı (ms)	params(M)
YOLOv8n-cls	224	69	88,3	0,31	2,7
YOLOv8s-cls	224	73,8	91,7	0,35	6,4
YOLOv8m-cls	224	76,8	93,5	0,62	17
YOLOv8l-cls	224	76,8	93,5	0,87	37,5
YOLOv8x-cls	224	79	94,6	1,01	57,4

Başarı Ölçütü

Derin öğrenme modellerinin performansını değerlendirmek, doğru sonuçlar elde edebilmek ve modeli optimize edebilmek için kritik öneme sahiptir. Eğitim başarı ölçütleri, modelin ne kadar iyi öğrenip genelleştirme yapabildiğini anlamaya yardımcı olur. Bu ölçütler, genellikle modelin eğitim ve test veri setlerindeki performansını kıyaslayarak belirlenmiştir. Eğitimin performansını belirlemek için bazı parametrelere sahip olmak gerekmektedir.

True Positives (TP): Doğru pozitif örnek sayısı.

True Negatives (TN): Doğru negatif örnek sayısı.

False Positives (FP): Yanlış pozitif örnek sayısı.

False Negatives (FN): Yanlış negatif örnek sayısı.

Doğruluk Oranı (Accuracy): Çalışmanın toplam doğru tahmin oranını göstermektedir ve 2.2 numaralı denklem ile hesaplanmaktadır.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.2)$$

Hassasiyet (Precision): Gerçek doğru tahminin, model tarafından yapılan toplam tahmine bölünmesi olarak tanımlanmaktadır.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.3)$$

Özgüllük (Recall): Gerçek pozitif örneklerin tüm pozitif örnekler içindeki oranını belirtmektedir. Özgüllük 2.4 denklemi ile hesaplanmaktadır.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.4)$$

F1 Puanı: Hassasiyet ve özgüllük arasındaki dengeyi sağlamak için kullanılır. F1 puanı 2.5 denklemi ile hesaplanmaktadır.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.5)$$

Konfüzyon Matrisi

Konfüzyon matrisi, sınıflandırma işlemlerinde bir modelin performansını değerlendirmek için kullanılan önemli bir matristir. Her sınıfın doğru ve yanlış sınıflandırılma sayılarını göstererek modelin nerede başarılı ve nerede hatalı olduğunu detaylı bir şekilde analiz edilmesine olanak tanımıştır.

Konfüzyon matrisi Şekil 2. 14'te verilmiştir.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Şekil 2. 14. Konfüzyon matrisi

Belirtilen metrikler, eğitimin genel performansını çok yönlü bir şekilde değerlendirmek için kullanılmıştır. Bu metrikler, her bir sınıfın doğruluğunu, hassasiyetini, özgüllüğünü ve F1 puanını değerlendirerek modelin güçlü ve zayıf yönlerini daha detaylı bir şekilde ortaya koymuştur.

ÖNİŞLEME

Veriler, derin öğrenme mimarisi içerisinde işlenmeden önce ön işleme adımlarından geçmektedir. Ön işleme, veri analizinin ve modellemenin temel bir parçasıdır. Verinin güvenilirliğini, kalitesini ve analiz edilebilirliğini artırarak daha iyi sonuçlar elde edilmesini sağlamaktadır.

Ses Ön İşleme

- **Sessiz Kısımları Kaldırma**

Konuşma ses verilerinde sessiz kısımlar olabilmektedir. Bu kısımlar bilgi taşımadıkları için önemsizdir. Bu kısımları tespit etmek ve atmak amacıyla “Sıfır Geçiş Hızı (ZCR)” ve “Kısa Süreli Enerji (STE)” gibi teknikler kullanılmaktadır. ZCR, bir ses sinyalinin belirli bir aralıkta genliğinin sıfır noktasından geçme sayısını ölçmektedir [8].

$$ZCR = \frac{1}{N-1} \sum_{n=1}^{N-1} | \text{sgn}(x[n]) - \text{sgn}(x[n-1]) | \quad (2.6)$$

2.6 numaralı denklem ile hesaplanır. N çerçeve uzunluğu, x[n] sinyalin n. örnek değeri, sgn(x) ise x değerinin işaret fonksiyonudur.

STE, bir ses sinyalinin belirli bir zaman aralığındaki toplam enerjisini ölçmektedir.

$$E_n = \frac{1}{N-1} \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \quad (2.7)$$

- **Ön Vurgu**

Ses sinyalinin frekans spektrumunu dengelemek için kullanılmaktadır. Bazı durumlarda, yüksek frekansların büyüklüğü düşük frekansların büyüklüğünden daha küçüktür. Bu sorun ön vurgu ile giderilmektedir. Ayrıca, sinyalin Fourier dönüşümü işleme sırasında sayısal sorunları ortadan kaldırır ve sinyal-gürültü oranını arttırmaktadır [24].

$$x'(n) = x(n) - \alpha x(n-1) \quad (2.8)$$

- **Gürültü Azaltma**

Gürültü azaltma, bir ses sinyalinin istenmeyen gürültüyü veya arka plan sesini ortadan kaldırmak veya azaltmak için kullanılan bir işlemdir. Bu işlem, ses kalitesini artırmaya ve daha net, daha temiz bir ses elde etmeye yardımcı olmaktadır. Ses verisinde arka plandaki gürültüyü bastırmak için zaman ve frekansta ekseninde filtre türleri ve adaptif filtre türleri kullanılmaktadır.

Frekans ekseninde filtre uygulamak için sinyalin Fourier dönüşümü alınarak (2.9) sinyal frekans eksenine taşınmaktadır. Frekans eksenindeki sinyale filtre uygulanır ve Ters Fourier dönüşümü (2.10) uygulanarak sinyal tekrardan zaman eksenine taşınmaktadır.

Fourier Dönüşümü;

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi jft} dt \quad (2.9)$$

Ters Fourier Dönüşümü;

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{2\pi jft} df \quad (2.10)$$

Gürültü azaltma tekniklerinden literatürde oldukça kullanılan filtrelerden biri ise Wiener Filtresidir [24]. Gürültünün istatistiksel özelliklerine dayanarak hesaplanan ortalama karesel hatanın minimizasyonu ölçütünü kullanarak gürültüyü an aza indirmektedir.

$$H(f) = \frac{S_s(f)}{S_s(f) + S_n(f)} \quad (2.11)$$

2.11 numaralı denklemde Wiener filtresinin transfer fonksiyonu verilmiştir. Burada $S_s(f)$ sinyalin güç spektral yoğunluğunu, $S_n(f)$ gürültünün güç spektral yoğunluğunu temsil etmektedir.

Gürültü azaltmak için zaman ekseninde M nokta kayan ortalama alıcı, üstel ağırlıklı ortalama kayan alıcı filtreler kullanılmaktadır. Bu filtreler, bir sinyalin zaman içinde değişen değerlerini belirli bir pencere boyunca ortalamasını alarak gürültüyü azaltmak ve sinyali pürüzsüzleştirmeyi amaçlamaktadır. Genellikler düşük frekanslı bileşenler korunurken yüksek frekanslı gürültü bastırılmış olur.

M nokta kayan ortalama alıcı filtre;

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} x[n - k] \quad (2.12)$$

Üstel ağırlıklı filtre;

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} \alpha^{k+1} x[n - k] \quad (2.13)$$

- **Normalizasyon**

Ses sinyalinin hacmini standart bir seviyeye ayarlamak için kullanılmaktadır. Bu işlem, her sinyal değerinin karşılaştırılabilir bir hacim seviyesine sahip olmasını sağlamaktadır. En yüksek hacim değeri belirlenmekte ve diğer değerler bu hacim değerine bölünerek verilerin hacmi belirli bir standarta getirilmektedir.

$$x_{\max} = \max |x[n]| \quad (2.14)$$

$$x[n] = \frac{x[n]}{x_{\max}} \cdot L \quad (2.15)$$

L hedeflenen referans değeridir. (Genellikle 1 tercih edilir.)

- **Pencereleme**

Pencereleme işlemi sinyal işlemede sıklıkla kullanılan bir ön işleme tekniğidir. Temel olarak, sinyalin belirli bir zaman aralığında parçalara ayrılması ve her bir parçanın yalnızca belirli bir zaman aralığında etkili olmasını sağlamaktadır. Sinyale pencere fonksiyonu uygulanmasının en büyük nedeni FFT tarafından oluşturulan ileriye dönüklüğü dengelemek ve spektral sızıntıyı hafifletmektir. Çeşitli pencereleme yöntemleri vardır. Dikdörtgen pencere, Hamming pencere, Hanning pencere, Kaiser pencere, Blackman penceresi [25]. Literatürde en çok kullanılan hamming penceresinin matematiksel fonksiyonu verilmiştir.

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right), 0 \leq n \leq N - 1, n \text{ pencere uzunluğudur.} \quad (2.16)$$

- **Görüntü Ön İşleme**

Görüntü ön işleme, ırk sınıflandırması, yüz tanıma, duygu tespiti ve yüz takibi gibi görevlerin başarısı için kritik bir ilk adımdır. Görüntü ön işleme, görüntü kalitesini

artırmak, istenmeyen özellikleri azaltmak ve önemli özellikleri vurgulamak amacıyla uygulanmaktadır. Literatürde yapılan araştırmalarda yüz ön işleme aşamaları aşağıdaki adımlardan oluşmaktadır.

- **Yüz Tespiti**

Yüz tespiti, bir görüntüdeki yüzleri belirleme işlemidir. Bu adım, yüz tanıma, duygu tespiti ve diğer yüz tabanlı işlemler için temel oluşturmaktadır. Öne çıkan klasik çalışmalar arasında Haar Cascades sınıflandırıcısı ve Histogram of Oriented Gradients (HOG) ve ardından gelen Destek Vektör Makineleri (SVM) bulunmaktadır [31][32].

Günümüzde, yüz tespiti için kullanılan modern yöntemler, derin öğrenme ve evrimsel sinir ağları (CNN) gibi daha gelişmiş tekniklere dayanmaktadır. Bu yöntemler, yüksek doğruluk ve hızlı işlem süreleri sunarak klasik yöntemlerin önüne geçmiştir. Bazı popüler kütüphaneler şunlardır:

OpenCV: Açık kaynaklı bir bilgisayarla görme kütüphanesidir ve çeşitli yüz tespit algoritmaları içermektedir. OpenCV, hem Haar Cascades hem de HOG + SVM yöntemlerini desteklemekte ve ayrıca derin öğrenme tabanlı yüz tespit yöntemlerine de sahiptir [33].

Dlib: Dlib, C++ tabanlı bir makine öğrenme kütüphanesidir ve yüz tespiti için son derece etkili araçlar sunmaktadır. Dlib'in HOG tabanlı yüz tespit yöntemi ve derin öğrenme tabanlı yüz tespit yöntemi, yüz tespitinde yüksek performans göstermektedir [33].

Face Recognition: Python tabanlı bu kütüphane, yüz tanıma ve tespit işlemlerini kolayca gerçekleştirmek için kullanılmaktadır. Dlib kütüphanesi üzerine inşa edilmiştir. Doğru ve hızlı yüz tespiti sağlar [33].

- **Yüz Kırpma**

Yüz tespiti sonrasında, algılanan yüz bölgesinin çevresindeki gereksiz kısımlar kaldırılarak yalnızca yüz kısmı kırpılmaktadır. Bu işlem, yüz tanıma ve öznitelik çıkarma süreçlerini daha verimli hale getirmektedir. Yüz kırpma işlemi, algılanan yüz bölgesinin sınırları belirlenerek gerçekleştirilmektedir.

- **Yeniden Boyutlandırma**

Yüz kırpma işleminden sonra, görüntülerin belirli bir boyuta yeniden ölçeklendirilmesi gerekmektedir. Bu adım, yüz tanıma modellerinin giriş gereksinimlerine uyum sağlamak için yapılmaktadır. Yeniden boyutlandırma işlemi, tüm görüntülerin aynı boyutlarda olmasını ve bu sayede karşılaştırılabilir ve tutarlı özelliklerin çıkarılmasını sağlamaktadır.

- **Normalizasyon**

Normalizasyon, veriyi 0 ile 1 arasında bir aralığa getirerek, optimizasyon sırasında gradyanın hesaplanmasını kolaylaştırmaktadır. Bu durumda, giriş görüntüsü piksel olarak temsil edilmekte ve her piksel, 0 ile 255 arasında değişen RGB (kırmızı, yeşil ve mavi) değerlerini içermektedir. Normalizasyon yapmak için, her pikselin her RGB değeri 255'e bölünerek nihai değerleri hesaplanmaktadır.

$$X_{\text{normalized}} = \frac{X}{255} \quad (2.17)$$

- **Gri Tonlama**

Gri tonlama, renkli görüntülerin gri ölçekli görüntülere dönüştürülmesidir. Bu işlem, görüntü işlemeyi basitleştirmekte ve hesaplama maliyetlerini düşürmektedir. Gri tonlama, her pikselin parlaklık değeri kullanılarak gerçekleştirilir ve bu sayede renk bilgisi yerine yalnızca yoğunluk bilgisi kullanılır.

2.2. Tasarım (Yöntemin Problemin Çözümünde Kullanımı)

Çalışmada ses ve görüntü verileri için kullanılan yöntemler farklılık göstermektedir. Bu sebepten dolayı ses ve görüntü verileri ayrı ayrı ele alınıp işlenmiştir. Bu bölümde dil ve ırk için uygulanan yöntemler ayrı ayrı anlatılacaktır.

DİL

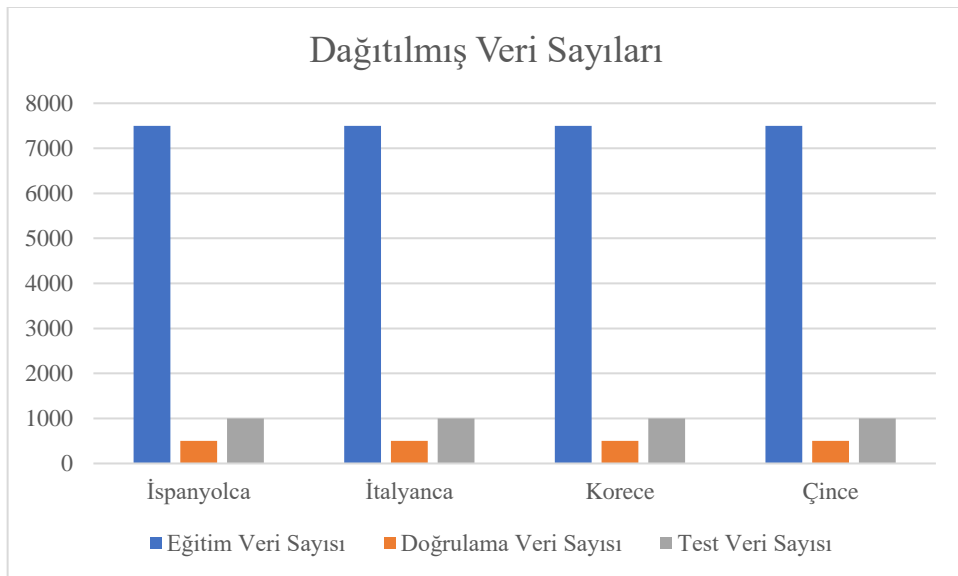
Dil Veri Seti

Farklı dilleri konuşan insanların konuşma seslerinden oluşan açık kaynaklı veri setleri bulunmaktadır. Bunlardan birkaçı, “Common voice by Mozilla”, “VoxForge”, “LibriSpeech”, “Tatoeba”, “VoxLingua107”dir.

Çalışmada dil tanıma sistemi için kullanılacak veri seti çeşitli dillerden kullanıcı tarafından gönderilen ses kliplerinden oluşan açık kaynaklı bir derleme olan VoxLingua107’dir [6]. Toplamda 107 farklı dilde ses dosyası bulunmaktadır.

VoxLingua107 veri setinden dil tanıma sisteminde kullanılacak diller indirilmiştir. İndirilen veri setine uygun ön işleme adımları uygulanmıştır. Veri setinde 4 sınıftan oluşan ve her sınıfta toplamda 9000 adet ses dosyalarının spektrograma dönüştürülmüş görüntüsü bulunmaktadır. Veri setindeki dil sınıfları “Korece”, “Çince”, “İspanyolca” ve “İtalyanca”dır. Spektrogramlar 224×224 boyutundadır. Bu spektrogramların %83’ü eğitim için %16’sı doğrulama ve test için kullanılmıştır.

Dil veri seti dağılımları Şekil 2. 15’te, Dil veri seti içerisindeki eğitim, doğrulama ve test veri sayısı Tablo 2. 2’de verilmiştir.



Şekil 2. 15. Dil veri seti dağılımları

Tablo 2. 2. Dil veri seti içerisindeki eğitim, doğrulama ve test veri sayısı

Sınıflar	Eğitim Veri Sayısı	Doğrulama Veri Sayısı	Test Veri Sayısı
İspanyolca	7500	500	1000
İtalyanca	7500	500	1000
Korece	7500	500	1000
Çince	7500	500	1000

Konuşulan Dil Ön İşleme Adımları

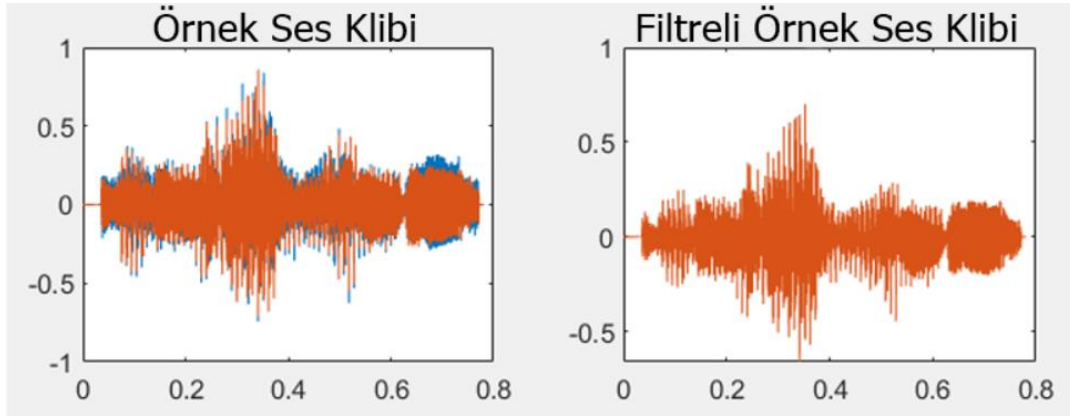
Açık kaynaklı bir derleme olan VoxLingua107 veri setinden İtalyanca, İspanyolca, Korece ve Çince dil sınıflarını seçilmiştir. Seçilen sınıfları içeren bu veri setine ön işleme adımları uygulanıp çalışmaya özel veri seti oluşturulmuştur. Özel veri setinde her bir dil sınıfı toplamda 9000 adet veri içermektedir. Ses verilerine uygulanan ön işleme adımları sırayla verilmiştir.

1. Gürültü Temizleme

Veri setinde bulunan ses kliplerinde gürültüler olduğu tespit edilmiştir. Bu gürültüleri bastırmak için Dijital Sinyal İşleme 'de zaman ekseninde örneklenmiş sinyallerin gürültüsünü azaltmak için yaygın olarak kullanılan “M nokta kayan ortalama alıcı filtre” kullanılmıştır. M nokta kayan ortalama alıcı filtre her bir çıkış verisini üretmek için giriş verisinden M sayısınca noktanın ortalamasını almaktadır [26]. M nokta kayan ortalama alıcı filtrenin denklem formu 2.18 numaralı denklemde verilmiştir.

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n - k] \quad (2.18)$$

Veri setindeki gürültüleri azaltmak için M nokta kayan ortalama alıcı filtresinin verilere uygulanması amacıyla Python programlama dili içerisinde Numpy kütüphanesi kullanılmıştır. Verilen denkleme uygun filtre algoritması yazılmış ve her bir ses dosyasına uygulanmıştır. Filtresiz ve filtreli ses klibi grafiği Şekil 2. 16’da verilmiştir.



Şekil 2. 16. Filtresiz ve filtreli ses klibi

2. Boş Veri Silme

Hazır veri setleri içerisinde bulunan ses kliplerinde konuşulmayan kısımlar bulunmaktadır. Bu kısımları temizlemek amacıyla filtreden geçirilen ses kliplerinin veri değerlerinin içerisinde negatif ve pozitif sayılar bulunmaktadır. Bu nedenle veri değerlerinin karesi alınmış ve tüm değerler pozitif çevrilmiştir. Veri kaybı olmaması için eşik değeri 0.0000001 olarak belirlenmiştir. Belirlenen bu eşik değerinin altında kalan veriler silinmiştir. Bunun için Python programlama dili içerisinde Pydub kütüphanesi kullanılmıştır. Ses verilerinin değerlerini çıkaracak ve eşik değerinin altında kalan verileri silip geri kalan verilerle sesi örnekleyecek algoritma yazılmış ve ses veri setine uygulanmıştır.

3. 5 sn'lik Veri Oluşturma

Derin öğrenme modellerinde doğruluğun yüksek olması için etkenlerden birisi de verilerin aynı sayıda ve aynı boyutta olmasıdır. Bu amaçla ses verilerinin aynı uzunlukta olması istenmiştir. Açık kaynaklı veri seti olan VoxLingua107 veri seti içerisindeki sesler farklı uzunluktadır. Sesler daha iyi bir öğrenme oranı için aynı uzunluğa getirilmiştir. 5 sn'den uzun sesleri 5 sn'lik klipler halinde kırpan algoritma Python programlama dili içerisinde Pydub kütüphanesi kullanarak yazılmıştır. Bu algortmada 5 sn altındaki sesler silinmiş, 5 sn üstündeki sesler 5'er saniyelik klipler halinde kırılmıştır.

4. Normalizasyon

Ses normalizasyonu ses sinyallerinin genliklerin ayarlanmasıdır. Ses sinyalleri belirli bir aralıkta ölçeklenmiştir. Ses normalizasyonu yapılırken dinamik aralığın korunması gerekmektedir. Dinamik aralık, ses sinyalinin en yüksek ve en düşük genliği arasındaki farkı ifade etmektedir. Bu aralığı korumak veri bozulmasını engellemek için önemlidir. Normalizasyon işlemi 2.19 numaralı denklemde verildiği şekilde yapılmaktadır. Bu denkleme uygun olarak ses verilerine normalizasyon işlemi uygulamak için Python programlama dili içerisinde Numpy ve Librosa kütüphaneleri kullanılarak uygun algoritma yazılmış ve ses verilerine uygulanmıştır.

$$x = \frac{\text{ses} - \min(\text{ses})}{\max(\text{ses}) - \min(\text{ses})} \quad (2.19)$$

5. Spektrograma Dönüştürme

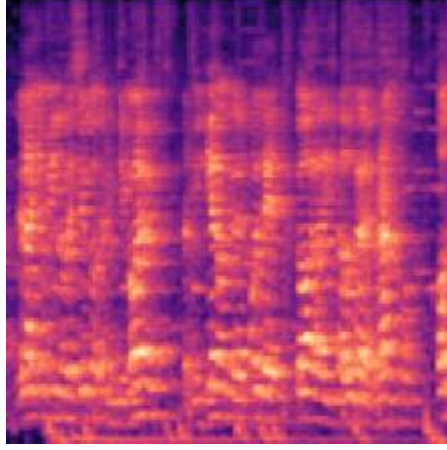
Spektrogram, bir sinyalde mevcut olan frekansların görselleştiren grafik türüdür. X eksenini zamanı, y eksenini frekans temsil etmektedir. Ses verilerinin spektrograma dönüştürülmesi ses sinyalinin zaman boyunca nasıl değiştiğini ve hangi frekanslarda hangi genlikte olduğunu anlamak için kullanılmaktadır. Spektrogram iki boyutlu olmasına rağmen frekansın genliğini gösteren bir ısı haritası olarak temsil edilen üçüncü bir boyutu da içermektedir. Ses verisinin bir bölümüne Hızlı Fourier Dönüşümü (FFT) (2.20 numaralı denklem) uygulanarak belirli bir an için frekans spektrumunun büyüklüğü hesaplanmaktadır. FFT verileri birleştirilerek frekansların zaman içerisinde nasıl değiştiğini görmemizi sağlayan spektrogramlar oluşturulmuştur. Ses verilerinde spektrogramlar oluşturulurken insan kulağının ses frekanslarındaki değişimi gösteren bir ölçek olan mel ölçeğine göre (2.21 numaralı denklem) spektrogramlar oluşturulmuştur. Çalışmada Python programlama dili içerisinde Numpy, Librosa ve Matplotlib kütüphaneleri kullanılarak 5 saniyelik ses verileri mel ölçeğine uygun olarak 224×224 piksel boyutunda spektrogramlara dönüştürülmektedir. Spektrograma dönüştürülmüş örnek ses klibi Şekil 2. 17’de verilmiştir.

Fourier Dönüşümü;

$$X(k) = \sum_{n=0}^{N-1} (x_n \cdot e^{-2\pi jkn/N}), \quad k = 0, 1, 2, \dots, N-1 \quad (2.20)$$

Mel Ölçeği;

$$f(mel) = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (2.21)$$



Şekil 2. 17. Spektrograma dönüştürülmüş örnek ses klibi

Dil Tanıma Modelleri

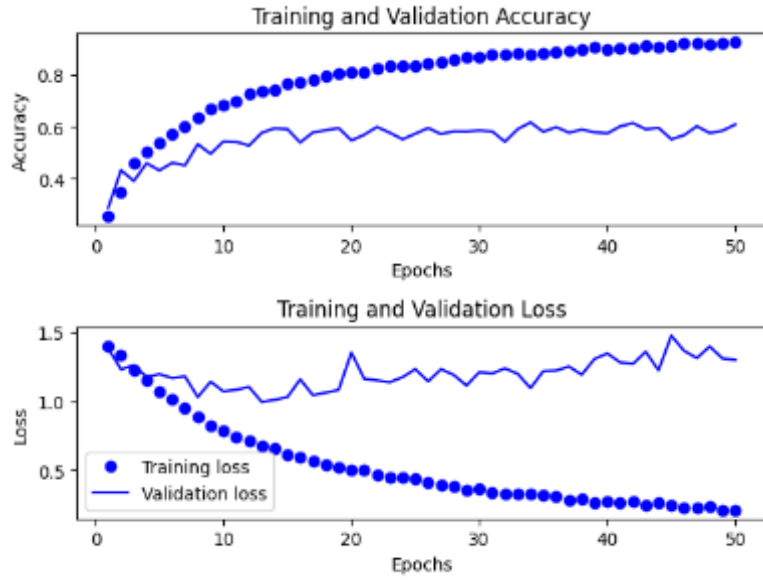
Dil tanıma sisteminin eğitimi için ses verilerine ön işleme adımları uygulanmıştır. Ön işleme sonrasında ses klipleri 224×224 piksel boyutunda spektrogramlara dönüştürülmüştür. Eğitim işlemi ses klipleri üzerinden değil ses kliplerinin spektrogram görüntüsünden yapılmıştır.

Çalışmada konuşulan dil tanıma sisteminin eğitimi için farklı CNN mimarileri tercih edilmiştir. Bunlar, mimarisi tasarlanan CNN yapısı, hazır CNN mimarileri ve YOLOv8-cls mimarisidir.

Dil Tanıma İçin Tasarlanan CNN Algoritması

Dil tanıma sistemi için CNN mimarisi tasarlanmıştır. Tasarlanan CNN katmanları Şekil 2. 19'da verilmiştir. Tasarlanan CNN mimarisinde katmanlar, hiper parametreler, kullanılan fonksiyonlar değiştirilerek en yüksek doğruluk oranı bulunmaya çalışılmıştır. Ancak istenilen doğruluk oranı elde edilememiştir. Tasarlanan CNN mimarisi ile eğitim yapıldığında doğruluk oranı 0,57 olarak elde edilmiştir. Doğruluk oranı düşük çıktığı için bu CNN mimarisi tasarlanan dil tanıma sisteminde kullanılmayacaktır. Eğitim sonucu oluşturulan doğruluk ve kayıp grafiği Şekil 2.

18’de, tasarlanan CNN mimarisinde kullanılan hiper parametreler Tablo 2. 3’te verilmiştir.



Şekil 2. 18. Dil tanıma sistemi için tasarlanan CNN mimarisinin eğitim sonuç grafiği

Tablo 2. 3. Dil tanıma sistemi için tasarlanan CNN mimarisinde kullanılan hiper parametreler

Optimizer	Öğrenme Oranı	Batch Boyutu	Epok
Adamax	0.0001	64	50

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 224, 224, 32)	896
conv2d_9 (Conv2D)	(None, 224, 224, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_10 (Conv2D)	(None, 112, 112, 64)	18496
conv2d_11 (Conv2D)	(None, 112, 112, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_12 (Conv2D)	(None, 56, 56, 128)	73856
conv2d_13 (Conv2D)	(None, 56, 56, 128)	147584
max_pooling2d_5 (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_14 (Conv2D)	(None, 28, 28, 256)	295168
conv2d_15 (Conv2D)	(None, 28, 28, 256)	590080
dropout_1 (Dropout)	(None, 28, 28, 256)	0
flatten_1 (Flatten)	(None, 200704)	0
dense_2 (Dense)	(None, 256)	51380480
dense_3 (Dense)	(None, 4)	1028
Total params: 52553764 (200.48 MB)		
Trainable params: 52553764 (200.48 MB)		
Non-trainable params: 0 (0.00 Byte)		

Şekil 2. 19. Dil tanıma sistemi için tasarlanan CNN katmanları

ResNet50 Mimarisi

Dil tanıma sistemi için ResNet50 mimarisi denenmiştir. ResNet50 mimarisinde son katmanlar ve hiper parametreler değiştirilerek en yüksek doğruluk oranı bulunmaya çalışılmıştır. Ancak yüksek doğruluk elde edilememiştir. Bu modelde yüksek doğruluk oranı elde edilemediği için dil tanıma sisteminde kullanılmamasına karar verilmiştir.

ResNet50 mimarisi ile eğitim yapıldığında doğruluk oranı 0,62 olarak elde edilmiştir. Eğitim sonucu oluşturulan doğruluk ve kayıp grafiği Şekil 2. 20’de, ResNet50 mimarisinde kullanılan hiper parametreler Tablo 2. 4’te verilmiştir.

Dil tanıma sistemi için kullanılan ResNet50 mimarisinin katmanları, ResNet50 mimarisinin katmanlarına ek olarak son katmanına sırayla “Flatten”, “Dropout (0.5)”, “Dense (512)”, “Dense (256)”, “Dense (128)” ve sınıflandırma katmanı olarak “Softmax” katmanı eklenmiştir.

Tablo 2. 4. Dil tanıma sistemi için ResNet50 mimarisinde kullanılan hiper parametreler

Optimizer	Öğrenme Oranı	Batch Boyutu	Epok
SGD	0.001	32	100



Şekil 2. 20. Dil tanıma sistemi için ResNet50 mimarisinin eğitim sonuç grafiği

YOLOv8-cls

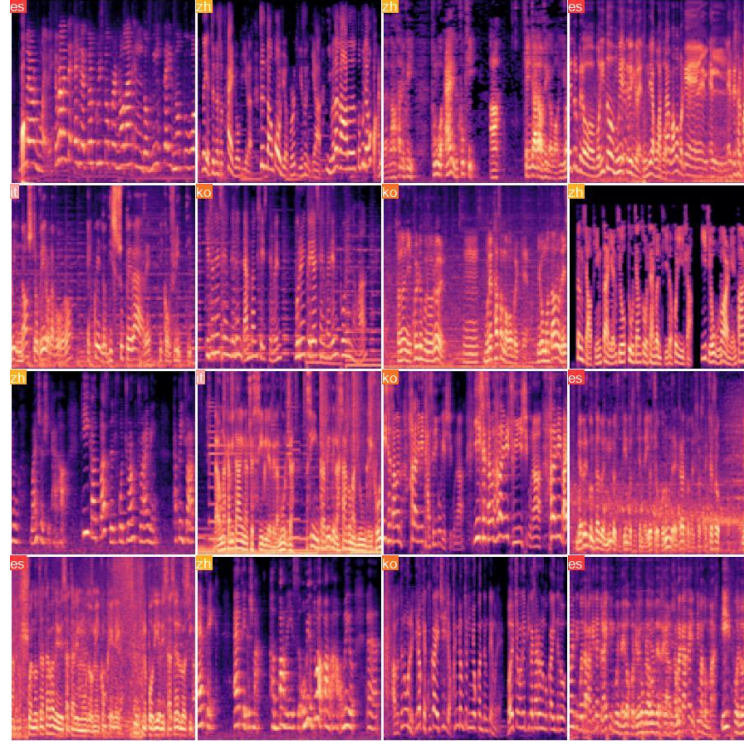
YOLOv8x-cls modeli, dil tanıma sistemi için test edilmiştir. Bu model, dil verisi eğitimi için kullanıldığında ilk denemede CNN ve ResNet50'ye kıyasla daha yüksek doğruluk sağlamıştır. Bu başarıya dayanarak, dil tanıma sistemlerinde YOLOv8x-cls kullanılması tercih edilmiştir.

YOLOv8x-cls ile yapılan denemelerde, eğitim ayarları ve hiper parametreler değiştirilerek en yüksek doğruluk elde edilmeye çalışılmıştır. Eğitim ayarları ve hiper parametreler değiştirilerek yapılan eğitimlerin sonuçları Tablo 2. 5'te verilmiştir. Bu eğitim sonuçlarına göre, en iyi sonuç KFG3 konfigürasyonunda %82,6 doğruluk oranıyla elde edilmiştir.

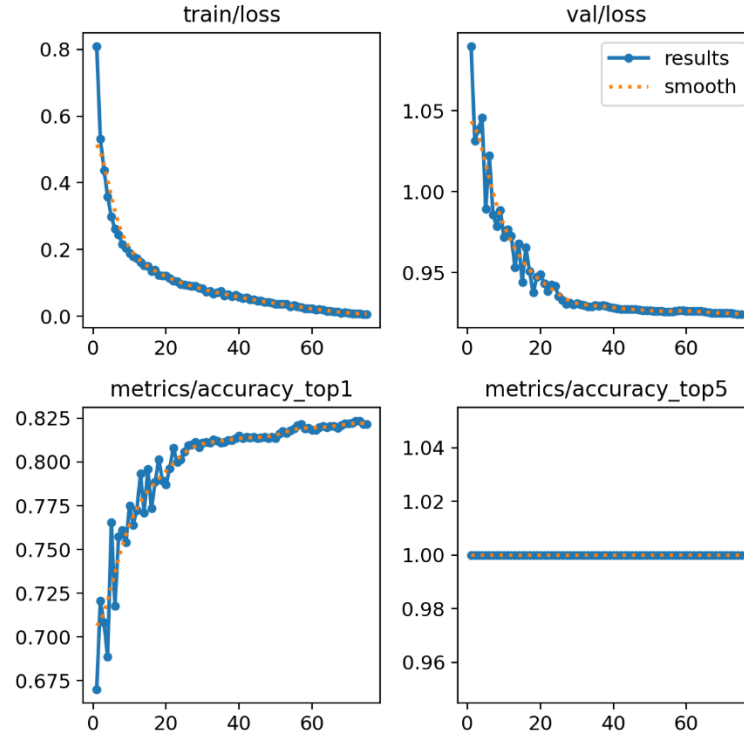
Tablo 2. 5. Dil tanıma sistemi için YOLOv8x-cls mimarisinde kullanılan farklı hiper parametreler ile eğitim sonuçları

Konfigürasyon	epok	imgsz	Batch	hsv_h	hsv_s	hsv_v	optimizer	lr0	Best Val Acc
KFG1	100	224	4	0,015	0,7	0,4	Auto'	0.001	74,8
KFG2	100	224	8	0,015	0,7	0,4	Auto'	0.001	75,9
KFG3	100	224	8	0	0	0	Adam	0.001	82,6
KFG4	75	224	16	0	0	0	SGD	0.01	75,1
KFG5	75	224	16	0	0	0	Adam	0.001	79,9
KFG6	75	224	64	0	0	0	Adam	0.001	81,7

Dil tanıma sistemi için en yüksek doğruluğu alan KFG3 konfigürasyonunun eğitim batch'i Şekil 2. 21'de gösterilmiştir. Eğitim sonuç grafiği ise Şekil 2. 22'de gösterilmiştir.



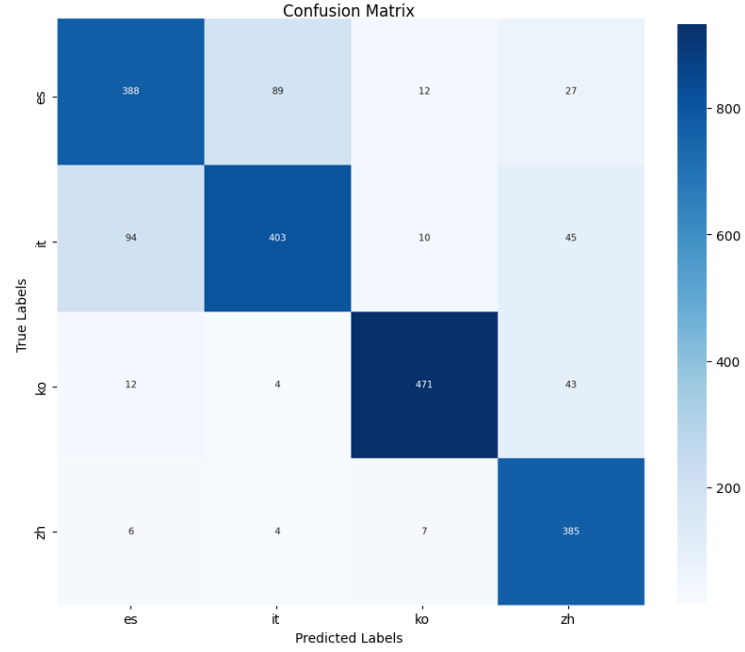
Şekil 2. 21. Dil tanıma sistemi KFG3 konfigürasyonunun eğitim batch'i



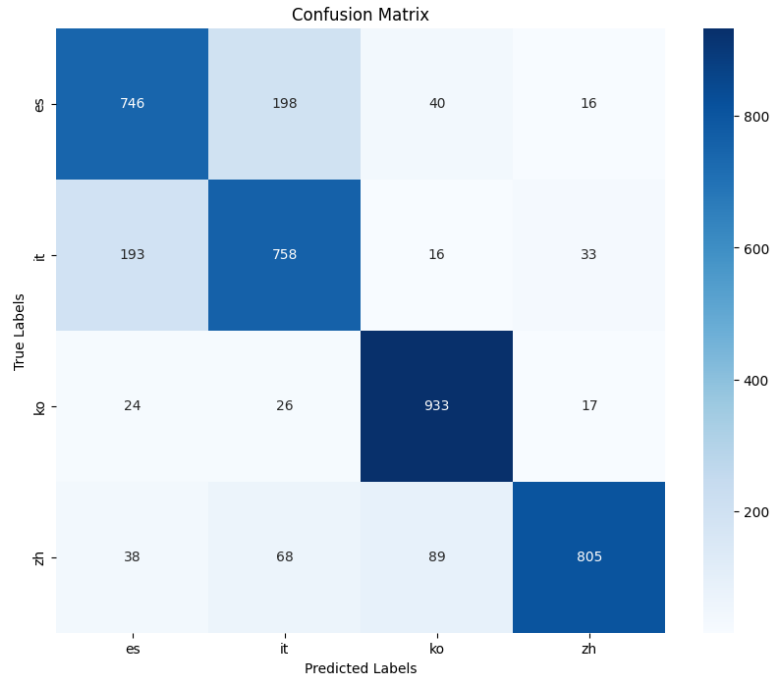
Şekil 2. 22. Dil tanıma sistemi KFG3 konfigürasyonunun eğitim sonuç grafiği

Eğitim esnasında yapılan testlerde en yüksek doğruluğu alan KFG3 konfigürasyonunun doğrulama veri seti ile yapılan testlerde %82,3 doğruluk oranına ulaşılmıştır. Şekil 2. 23'te yapılan doğrulama testinin konfüzyon matrisi verilmiştir.

Projede kullanılacak modelin ağırlık dosyası test seti ile test edilmiştir. Bu testlerde %81 doğruluk oranı elde edilmiştir. Yapılan test sonucunun konfüzyon matrisi Şekil 2. 24'te verilmiştir.



Şekil 2. 23. Dil tanıma sistemi KFG3 konfigürasyonunun doğrulama seti konfüzyon matrisi sonucu



Şekil 2. 24. Dil tanıma sistemi KFG3 konfigürasyonunun test seti konfüzyon matrisi sonucu

GÖRÜNTÜ

Görüntü Veri Seti

Farklı yüzlerden oluşan ırk, yaş ve cinsiyet verileri içeren veri setleri bulunmaktadır. Bunlardan birkaçı “VNFaces”, “FERET”, “VMER”, “UTK-Face” ve “FairFace”dir. Irk tanıma çalışmasında kullanılacak veri seti, Kärkkäinen ve Joo tarafından oluşturulan "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation" adlı çalışmadan gelmektedir. Bu veri seti, 1 ila 70 yaşları arasındaki farklı kişilerin ırk gruplarına ayrılmış 108,501 yüz görüntüsünden oluşmaktadır. Bu gruplar arasında Beyaz, Siyah, Latin, Doğu Asyalı, Güneydoğu Asyalı, Hintli ve Orta Doğulu bulunmaktadır. Bu veri seti, “2021 IEEE Winter Conference on Applications of Computer Vision (WACV)” kapsamında sunulmuş ve tanıtılmıştır [29].

Çalışmada sınıflar Latin, Doğu Asyalı ve diğerleri olarak 3’e ayrılmıştır. Veri setlerinde spesifik olarak “diğerleri” sınıfı olmadığı için çalışmada "diğerleri" sınıfında FairFace veri setindeki Beyaz, Siyah ve Hintli sınıfları kullanılmıştır. Çalışmada kullanılacak sınıfların FairFace veri setindeki toplam veri sayıları Tablo 2.6’da verilmiştir.

Tablo 2.6. FairFace veri seti toplam veri sayısı

Sınıflar	Toplam Veri Sayısı
Siyah	13789
Doğu Asyalı	13837
Hintli	13835
Latin	14990
Beyaz	18612
Tüm Sınıflar	75063

Görüntü Ön İşleme Adımları

Derin öğrenme çalışmalarında görüntü ön işleme önem arz etmektedir. Çalışmada açık kaynaklı FairFace veri setinden Latin, Doğu Asyalı, Beyaz, Siyah ve Hintli ırk sınıfları seçilmiştir. Seçilen bu sınıflara ön işleme adımları uygulanarak özel veri seti

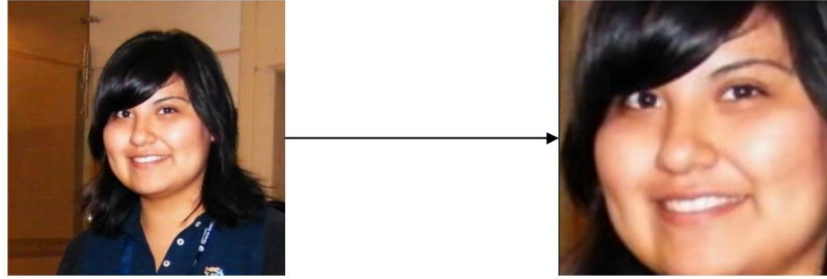
oluřturulmuřtur. Ses verilerine uygulanan 6n iřleme adımları sırayla Yüz ve G6z Tespiti, Kırpma, Yeniden Boyutlandırma, Normalizasyon ve Veri Seti B6lme iřlemleridir.

1. Yüz ve G6z Tespiti

Çalıřmada yüz ve g6z tespiti iřlemleri Python programlama dili ierisinde OpenCV kütüphanesi kullanılarak frontal yüz ve g6z detekt6rleri ile yapılmıřtır. İki adetden az g6z bulunan fotoğraflar bir sonraki ařamaya geememiřtir.

2. Kırpma

Yüz ve g6z tespiti ařamasında tespit edilen yüzler, frontal yüz detekt6r6n belirlediėi çereve sınırları ile kırpılmıřtır. Kırpılmıř yüz verisi 6rneėi řekil 2. 25'te verilmiřtir.



řekil 2. 25. Kırpılmıř yüz verisi 6rneėi

3. Yeniden Boyutlandırma

Kesilen yüz fotoğraflarının boyutları, YOLOv8 modeline uyumlu hale getirilmesi iin yeniden boyutlandırılmıřtır. T6m yüz g6r6nt6leri 224×224 piksel boyutlarına ayarlanmıřtır, b6ylece modelin girdi gereksinimlerine uygun hale getirilmıřtir.

4. Normalizasyon

Normalizasyon iřleminde her pikselin deėeri, 0-255 aralıėındaki piksel deėerinin 255'e b6l6nmesiyle 0-1 aralıėına getirilmıřtir. Ařaėıda bu iřlem iin kullanılan Python kodu 2.22 numaralı denklemde verilmiřtir.

$$\text{normalized_img} = \text{img.astype('float32')} / 255.0 \quad (2.22)$$

Ön işleme adımlarının sonucunda, ırk veri setindeki fotoğrafların yarısı elenmiştir. Kalan veri sayıları, Tablo 2. 7’de gösterilmiştir.

Tablo 2. 7. Çalışmada kullanılacak ırk veri seti veri adetleri

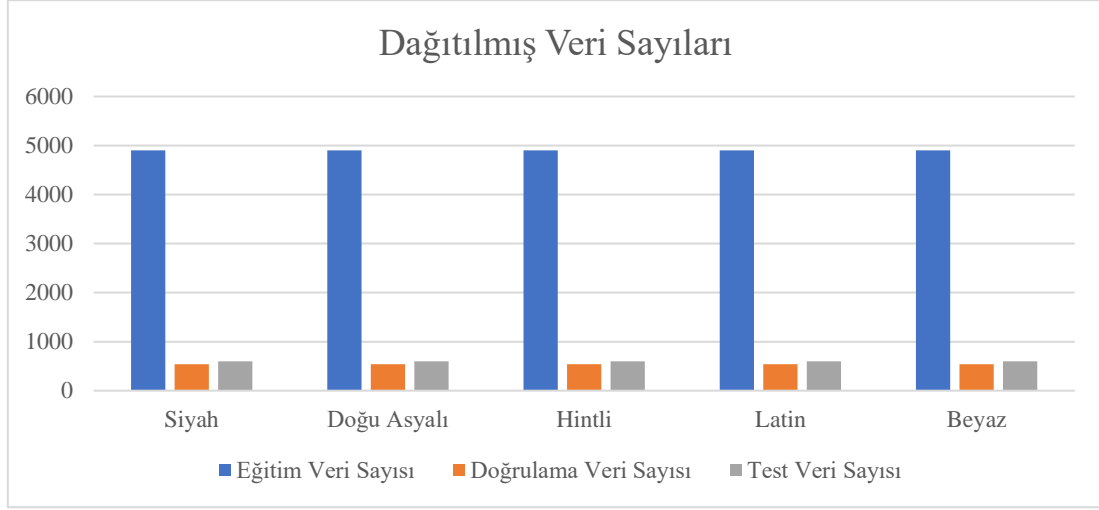
Sınıflar	Ön işlenmiş Toplam Veri Adedi
Siyah	6040
Doğu Asyalı	6040
Hintli	6040
Latin	6040
Beyaz	6040
Tüm Sınıflar	30200

5. Veri Seti Bölme

Ön işleme aşamasının son adımında veri seti, eğitim, doğrulama ve test olarak ayrılmıştır. Oluşturulan veri setlerinin dağılımı Şekil 2. 26 ve Tablo 2. 8’de gösterilmiştir.

Tablo 2. 8. Irk veri seti içerisindeki eğitim, doğrulama ve test veri sayısı

Sınıflar	Eğitim Veri Sayısı	Doğrulama Veri Sayısı	Test Veri Sayısı
Siyah	4900	540	600
Doğu Asyalı	4900	540	600
Hintli	4900	540	600
Latin	4900	540	600
Beyaz	4900	540	600
Tüm Sınıflar	24500	2700	3000



Şekil 2. 26. Irk veri seti dağılımları

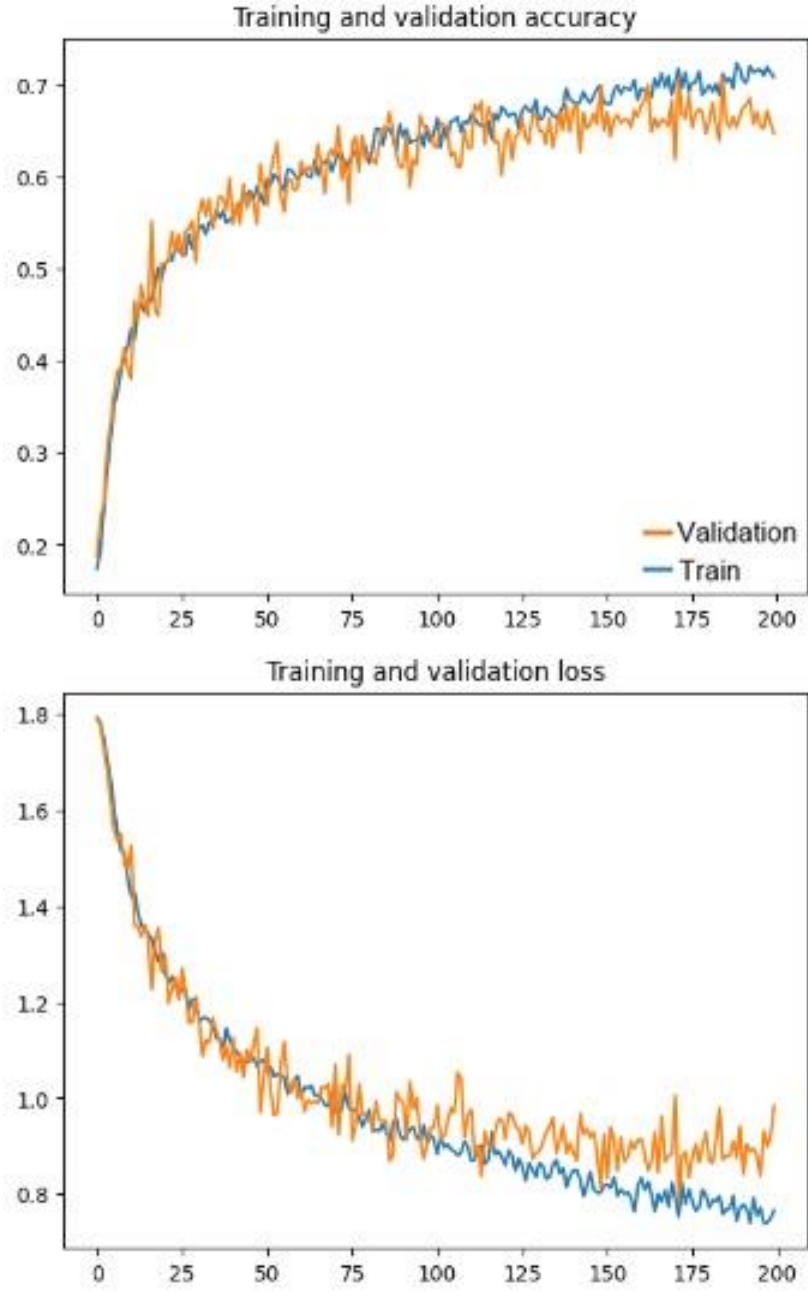
İrk Tanıma Modeli

İrk tanıma sisteminin eğitimi için ırk verilerine ön işleme adımları uygulanmıştır. Ön işleme sonrasında veri setindeki uygun olmayan veriler atılmış ve veriler 224×224 piksel boyutuna ölçeklendirilmiştir.

Çalışmada ırk tanıma sisteminin eğitimi için farklı CNN mimarileri tercih edilmiştir. Bunlar, mimarisi tasarlanan CNN yapısı, hazır CNN mimarileri ve YOLOv8-clc mimarisidir.

İrk Tanıma İçin Tasarlanan CNN Algoritması

İrk tanıma sistemi için CNN mimarisi tasarlanmıştır. Tasarlanan CNN katmanları Şekil 2. 28'de verilmiştir. Tasarlanan CNN mimarisinde katmanlar, hiper parametreler, kullanılan fonksiyonlar değiştirilerek en yüksek doğruluk oranı bulunmaya çalışılmıştır. Ancak istenilen doğruluk oranı elde edilememiştir. Tasarlanan CNN mimarisi ile eğitim yapıldığında doğruluk oranı 0,65 olarak elde edilmiştir. Doğruluk sonucu düşük çıktığı için tasarlanan ırk tanıma sisteminde bu model sonucu kullanılmayacaktır. Eğitim sonucu oluşturulan doğruluk ve kayıp grafiği Şekil 2. 27'de tasarlanan CNN mimarisinde kullanılan hiper parametreler Tablo 2. 9'da verilmiştir.



Şekil 2. 27. Irk tanıma için tasarlanan CNN modelinin eğitim sonuç grafiği

Tablo 2. 9. Irk tanıma sistemi için tasarlanan CNN mimarisinde kullanılan hiper parametreler

Optimizer	Öğrenme Oranı	Batch Boyutu	Epok
Adamax	0.0005	20	200

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 224, 224, 16)	448
conv2d_25 (Conv2D)	(None, 224, 224, 16)	2320
max_pooling2d_12 (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_26 (Conv2D)	(None, 112, 112, 32)	4640
conv2d_27 (Conv2D)	(None, 112, 112, 32)	9248
max_pooling2d_13 (MaxPooling2D)	(None, 56, 56, 32)	0
conv2d_28 (Conv2D)	(None, 56, 56, 64)	18496
conv2d_29 (Conv2D)	(None, 56, 56, 64)	36928
max_pooling2d_14 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_30 (Conv2D)	(None, 28, 28, 128)	73856
conv2d_31 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_15 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_32 (Conv2D)	(None, 14, 14, 256)	295168
conv2d_33 (Conv2D)	(None, 14, 14, 256)	590080
max_pooling2d_16 (MaxPooling2D)	(None, 7, 7, 256)	0
dropout_3 (Dropout)	(None, 7, 7, 256)	0
flatten_3 (Flatten)	(None, 12544)	0
dense_6 (Dense)	(None, 256)	3211520
dense_7 (Dense)	(None, 6)	1542
=====		
Total params: 4391830 (16.75 MB)		
Trainable params: 4391830 (16.75 MB)		
Non-trainable params: 0 (0.00 Byte)		

Şekil 2. 28. Irk tanıma sistemi için tasarlanan CNN katmanları

GoogleNet

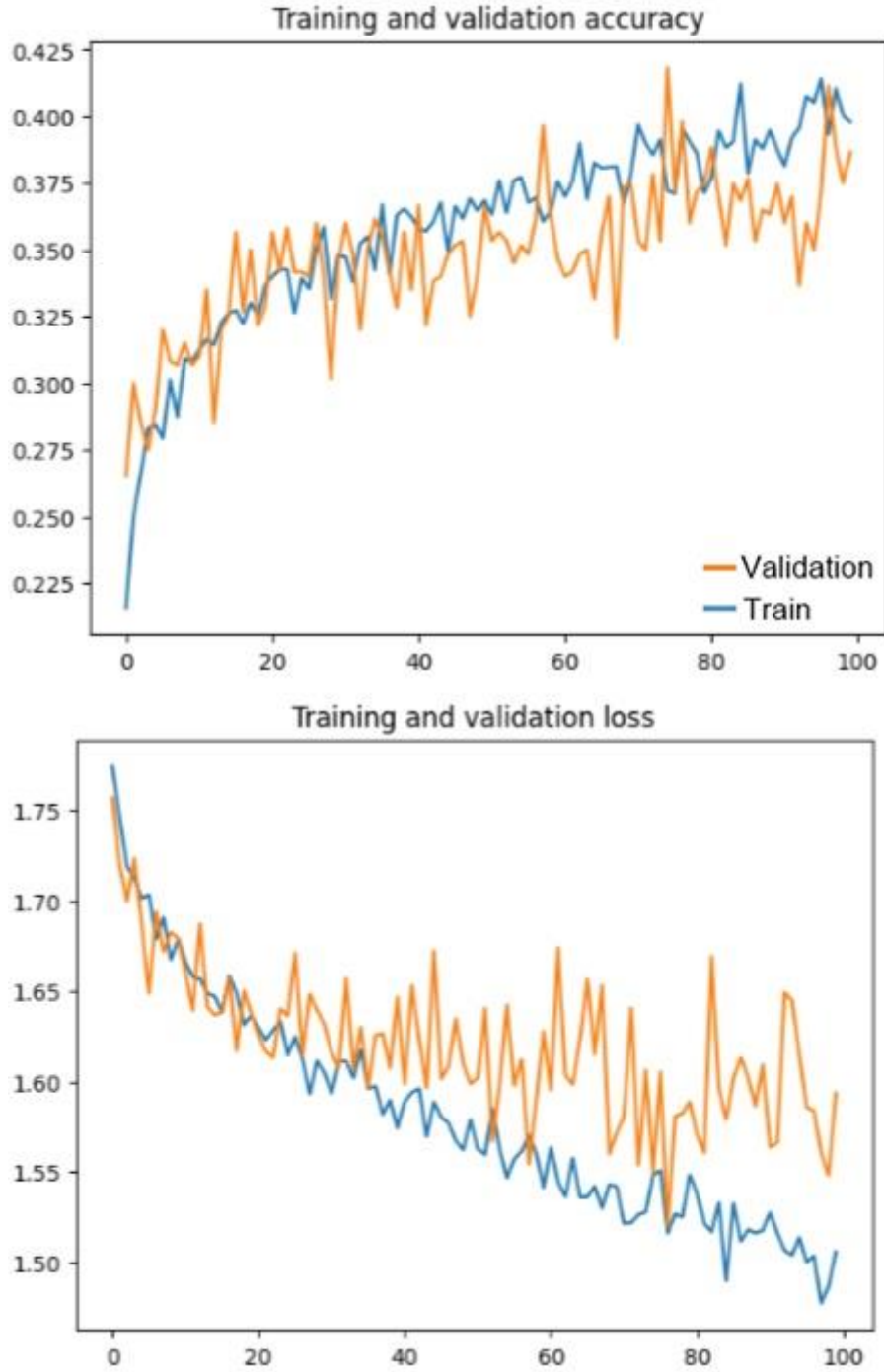
İrk tanıma sistemi için GoogleNet mimarisi denenmiştir. GoogleNet mimarisinde son katmanlar ve hiper parametreler değiştirilerek en yüksek doğruluk oranı bulunmaya çalışılmıştır. Ancak yüksek doğruluk elde edilememiştir. Bu modelde yüksek doğruluk oranı elde edilemediği için ırk tanıma sisteminde kullanılmamasına karar verilmiştir.

GoogleNet mimarisi ile eğitim yapıldığında doğruluk oranı 0,38 olarak elde edilmiştir. GoogleNet mimarisinde kullanılan hiper parametler Tablo 2. 10'da, eğitim sonucu oluşturulan doğruluk ve kayıp grafiği Şekil 2. 29'da verilmiştir.

Dil tanıma sistemi için kullanılan GoogleNet mimarisinin katmanları, GoogleNet mimarisinin katmanlarına ek olarak son katmanına sırayla “Flatten”, “Dense (512)”, Dropout (0.5)” ve sınıflandırma katmanı olarak “Softmax” katmanı eklenmiştir.

Tablo 2. 10. İrk tanıma sistemi için GoogleNet mimarisinde kullanılan hiper parametreler

Optimizer	Öğrenme Oranı	Batch Boyutu	Epok
Adam	0.0005	20	100



Şekil 2. 29. Irk tanıma sistemi için GoogleNet mimarisinin eğitim sonuç grafiği

VGG-16

İrk tanıma sistemi için VGG-16 mimarisi denenmiştir. VGG-16 mimarisinde son katmanlar ve hiper parametreler değiştirilerek en yüksek doğruluk oranı bulunmaya çalışılmıştır. Ancak yüksek doğruluk elde edilememiştir. Bu modelde yüksek doğruluk oranı elde edilemediği için ırk tanıma sisteminde kullanılmamasına karar verilmiştir.

İrk tanıma sistemi için VGG-16 mimarisinde kullanılan hiper parametreler Tablo 2. 11’de verilmiştir.

VGG-16 mimarisi ile eğitim yapıldığında doğruluk oranı 0,55 olarak elde edilmiştir. Eğitim sonucu oluşturulan doğruluk ve kayıp grafiği Şekil 2. 31’de verilmiştir.

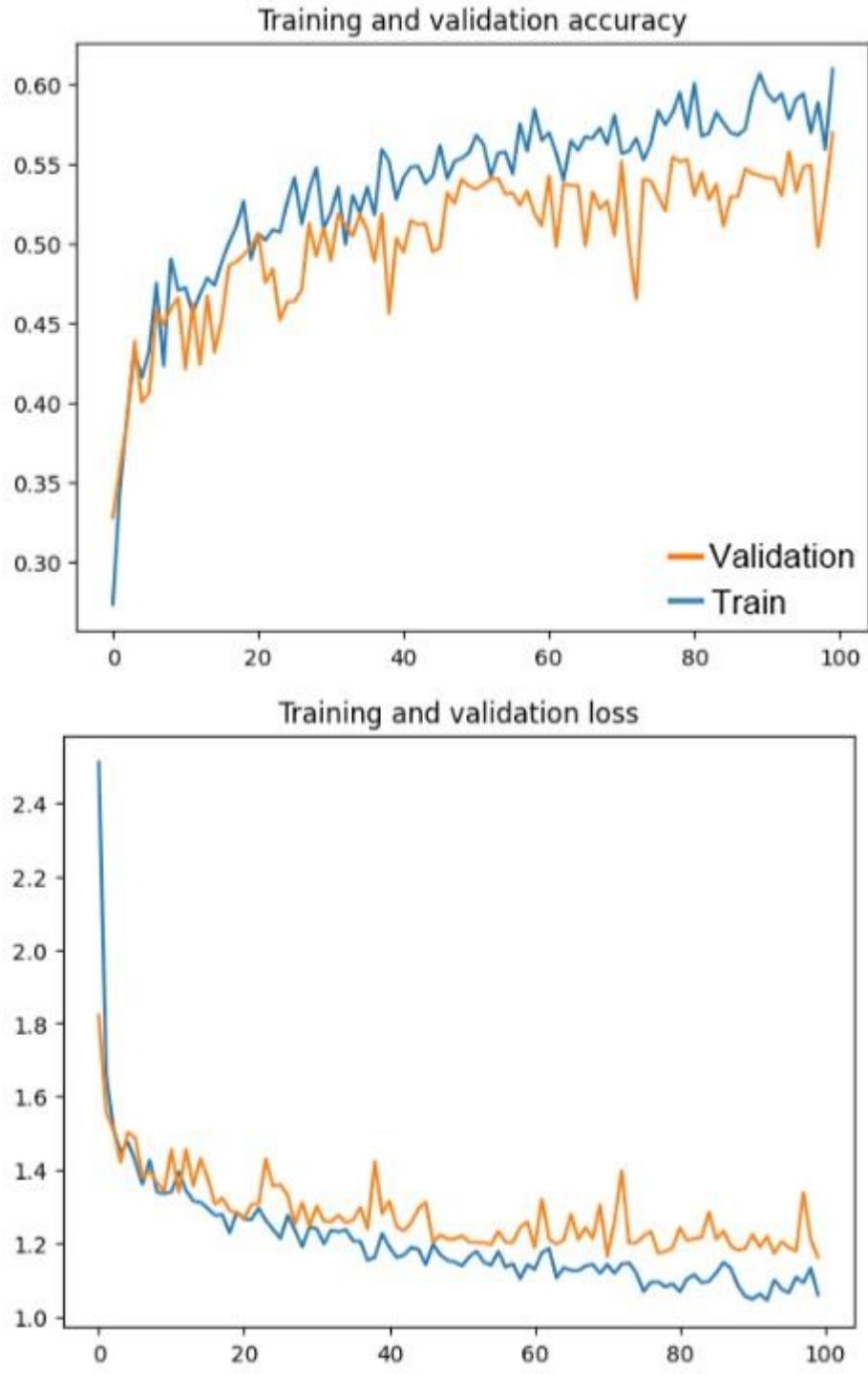
Dil tanıma sistemi için kullanılan VGG-16 mimarisinin katmanları Şekil 2. 30’da gösterilmiştir.

Tablo 2. 11. İrk tanıma sistemi için VGG-16 mimarisinde kullanılan hiper parametreler

Optimizer	Öğrenme Oranı	Batch Boyutu	Epok
Adam	0.0005	20	100

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 512)	12845568
dense_1 (Dense)	(None, 6)	3078
=====		
Total params: 27563334 (105.15 MB)		
Trainable params: 12848646 (49.01 MB)		
Non-trainable params: 14714688 (56.13 MB)		

Şekil 2. 30. Irk tanıma sistemi için kullanılan VGG-16 katmanları



Şekil 2. 31. Irk tanıma sistemi için VGG-16 mimarisinin eğitim sonuç grafiği

YOLOv8-clb

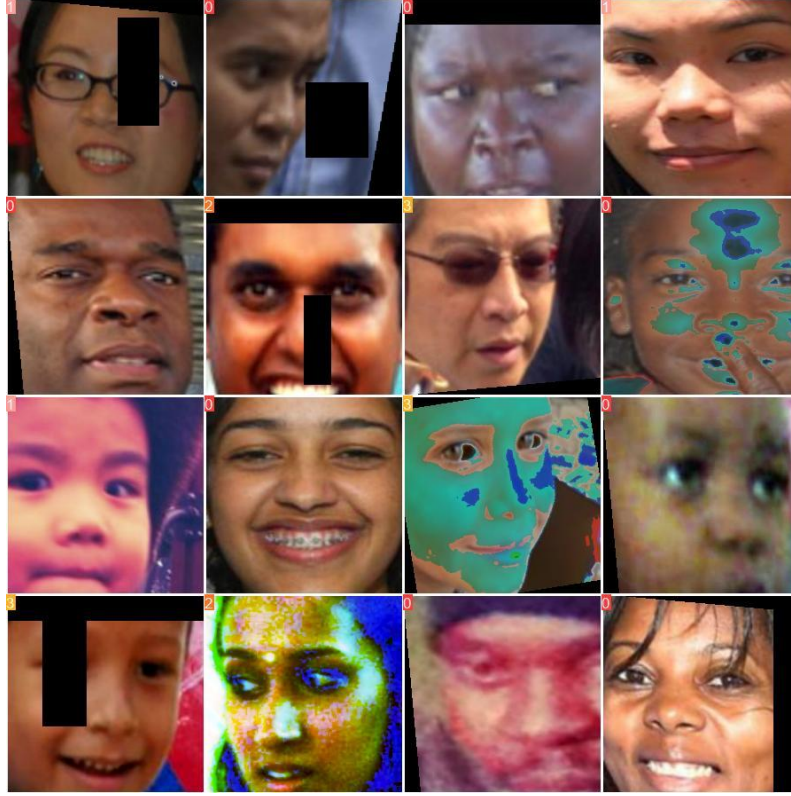
YOLOv8x-clb modeli, ırk tanıma sistemi için test edilmiştir. Bu model, ırk verisi eğitimi için kullanıldığında ilk denemede CNN, GoogleNet ve VGG-16'ya kıyasla daha yüksek doğruluk sağlamıştır. Bu başarıya dayanarak, ırk tanıma sistemlerinde YOLOv8x-clb tercih edilmiştir.

YOLOv8x-clb ile yapılan denemelerde, eğitim ayarları ve hiper parametreler değiştirilerek en yüksek doğruluk oranı elde edilmeye çalışılmıştır. Eğitim ayarları ve hiper parametreler değiştirilerek yapılan eğitimlerin sonuçları Tablo 2. 12'de verilmiştir. Bu eğitim sonuçlarına göre, en iyi sonuç KFG4 konfigürasyonunda %80,5 doğruluk oranıyla elde edilmiştir.

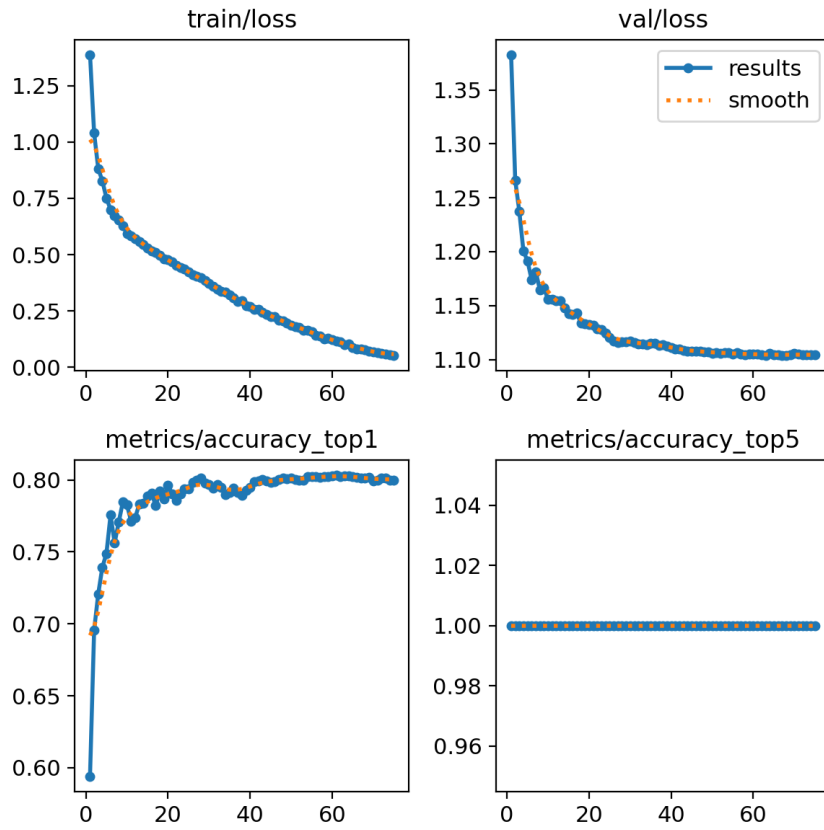
Tablo 2. 12. Irk tanıma sistemi için YOLOv8x-clb mimarisinde kullanılan farklı hiper parametreler ile eğitim sonuçları

Konfigürasyon	epochs	imgsz	batch	hsv_h	hsv_s	hsv_v	scale	optimizer	lr0	Acc
KFG1	75	224	32	0,015	0,7	0,4	0,5	SGD	0.01	79,1
KFG2	75	224	16	0,015	0,7	0,4	0,5	SGD	0.01	80
KFG3	75	224	8	0,015	0,7	0,4	0,5	SGD	0.01	79,8
KFG4	75	224	16	0,015	0,7	0,4	0	SGD	0.01	80,5
KFG5	75	224	16	0,015	0,7	0,7	0	Adam	0.01	80,4

Irak tanıma sistemi için en yüksek doğruluđu alan KFG4 konfigürasyonunun eğitim batch'i Şekil 2. 32'de gösterilmiştir. Eğitim sonuç grafiđi ise Şekil 2. 33'te gösterilmiştir.

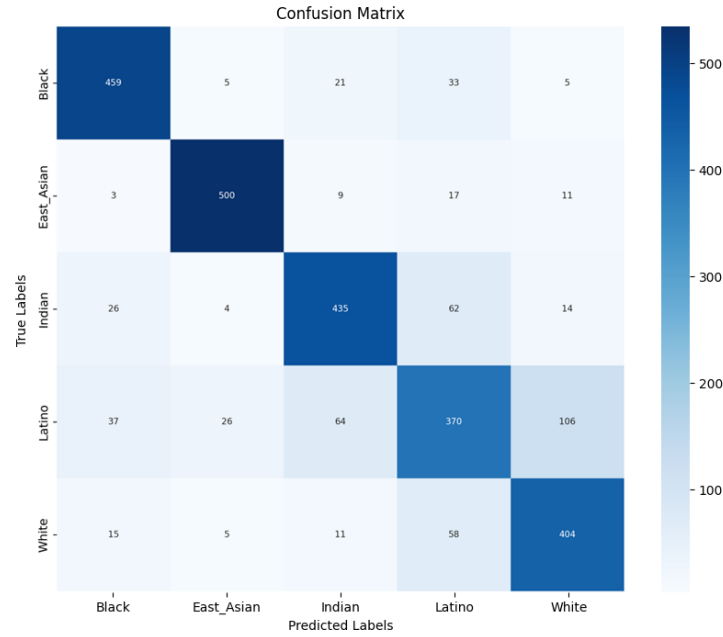


Şekil 2. 32. Irk tanıma sistemi KFG4 konfigürasyonunun eğitim batch'i

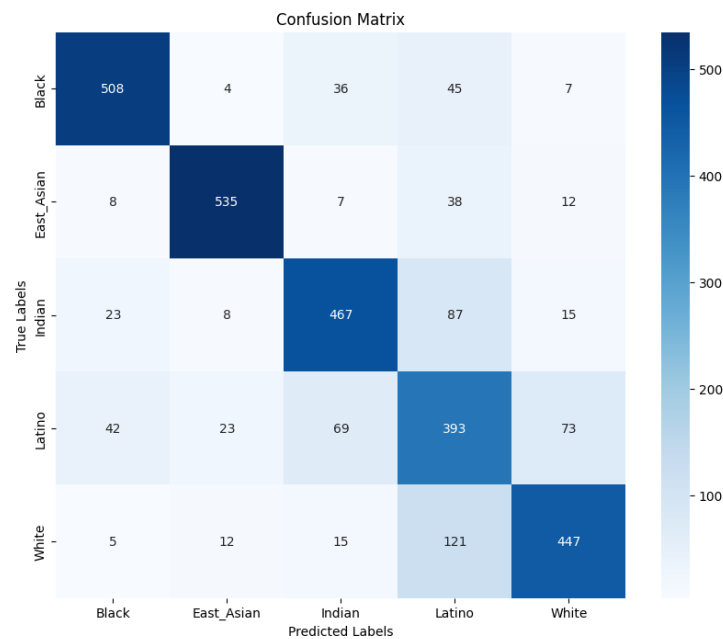


Şekil 2. 33. Irk tanıma sistemi KFG4 konfigürasyonunun eğitim sonuç grafiği

Eğitim esnasında yapılan testlerde en yüksek doğruluğu alan KFG4 konfigürasyonunun doğrulama veri seti ile yapılan testlerde %80,5 doğruluk oranına ulaşılmıştır. Şekil 2. 34'te yapılan doğrulama testinin konfüzyon matrisi bulunmaktadır. Projede kullanılacak modelin ağırlık dosyası test seti ile test edilmiştir. Bu testlerde %78,3 doğruluk oranı elde edilmiştir. Yapılan testin konfüzyon matrisi Şekil 2. 35'te yer almaktadır.



Şekil 2. 34. Irk tanıma sistemi KFG4 konfigürasyonunun doğrulama seti konfüzyon matrisi sonucu



Şekil 2. 35. Irk tanıma sistemi KFG4 konfigürasyonunun test seti konfüzyon matrisi sonucu

BÖLÜM 3. UYGULAMA ÇALIŞMALARI

Bu bölümde çalışmada gerçekleştirilmesi hedeflenen karmaşık bir sistem, süreç, cihaz, ürün veya uygulama (algoritma, arayüz veya donanım) açıklanmalı, deneysel çalışmalar ve tasarlanan uygulamaya ilişkin alt bileşenlere ait blok diyagramı mutlaka verilmelidir.

3.1. Uygulamada Kullanılan Araç ve Gereçler

- **Python Yazılım Dili**

Python, okunması ve öğrenilmesi kolay bir programlama dili olarak öne çıkmaktadır. Ücretsiz ve açık kaynaklı olan bu yazılım, kullanıcıların istedikleri gibi faydalanabilmesine olanak tanımaktadır. Hem okunabilir hem de yüksek performanslı kod yazımına imkân sağlamaktadır. Python, bilimsel alanlarda, özellikle makine öğrenimi ve veri bilimi konularında sıkça kullanılmaktadır.

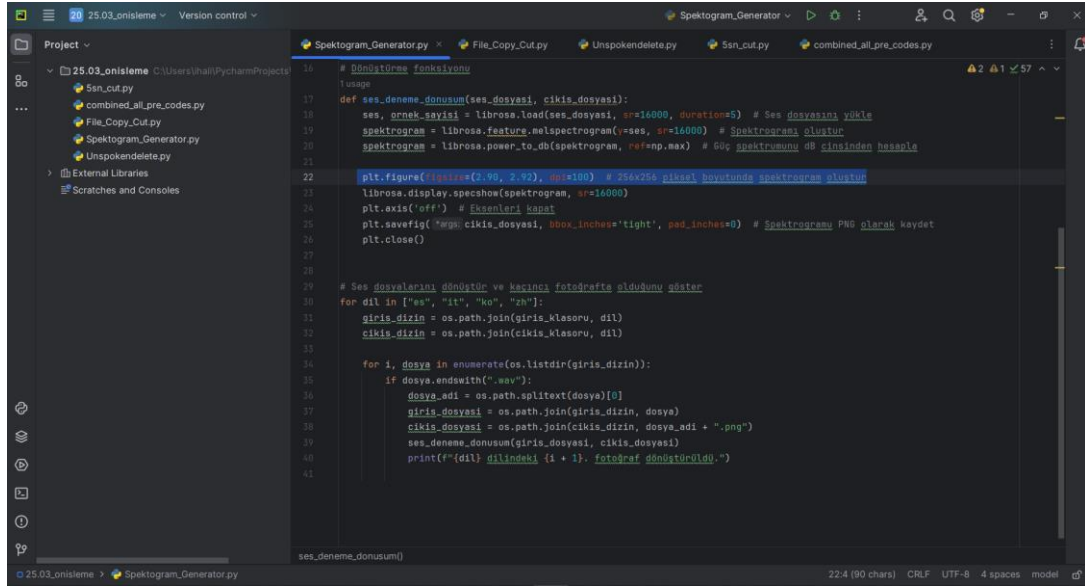
Çalışmada Python'un tercih edilmesi çeşitli sebeplere dayanmaktadır. İlk olarak Python, derin öğrenme ve yapay zekâ alanında kullanılmak üzere zengin bir kütüphaneye sahiptir. Özellikle TensorFlow, PyTorch, Keras, gibi kütüphaneler, derin öğrenme projelerinde sıklıkla kullanılmaktadır. Python'un geniş platform desteği sayesinde, Windows, Linux, MacOS ve diğer birçok işletim sistemine uyumlu olması, projelerin farklı işletim sistemlerinde çalıştırılmasını mümkün kılarak kullanıcı tabanını genişletmektedir. Ayrıca Python'un tamamen ücretsiz olması ve açık kaynaklı olması çalışmaya ekonomik avantaj sağlamıştır.

Python'un geniş ve zengin kütüphanesi kullanıcılara hazır modüller ve çeşitli araçlar sunarak yazılım geliştirme sürecini hızlandırmaktadır.

- **PyCharm IDE**

PyCharm IDE, JetBrains tarafından geliştirilen ve Python programlama dili için özel olarak tasarlanmış bir bütünleşmiş geliştirme ortamıdır (IDE). Python tabanlı projelerin geliştirilmesi için birçok özellik sunmakta ve geliştiricilere verimli bir şekilde çalışmalarını sağlamaktadır. Çalışmada Python dilini yazmak için PyCharm IDE ortamının tercih edilmesi çeşitli teknik ve pratik avantajlara dayanmaktadır.

Kodlama sürecini kolaylaştırmak için araçlar içermektedir. Otomatik kod tamamlama, hata ayıklama, kod analizi gibi özellikler, kullanıcının daha hızlı ve daha verimli bir şekilde kod yazmalarını sağlamaktadır. Ayrıca PyCharm, projelerin yönetimini kolaylaştırmaktadır. Bütünleşmiş proje ve dosya yönetimi özellikleri, projelerin düzenli ve organize bir şekilde tutulmasını sağlamaktadır. Kullanıcılar, projeler arasında kolayca geçiş yapabilmekte ve farklı dosyalar arasında hızlı bir şekilde gezinebilmektedir. PyCharm IDE arayüzü Şekil 3. 1'de verilmiştir-gösterilmiştir.



Şekil 3. 1. PyCharm IDE arayüzü

- **PyQt5 Arayüz Tasarım Kütüphanesi**

PyQt5, Python programlama dilini kullanarak masaüstü uygulamaları geliştirmek için kullanılan ve Qt Creator uygulama geliştirme çerçevesine bağlanmayı sağlayan güçlü bir arayüz tasarım kütüphanesidir. PyQt5, yüksek performanslı grafik arayüzler oluşturmak için geniş bir bileşen yelpazesi sunmaktadır. Bu bileşenler, kullanıcı dostu ve etkileşimli arayüzlerin geliştirilmesini sağlamaktadır. Yapılan çalışma kullanıcılara sunulurken kullanıcıların projeyi rahatça kullanabilmesi için arayüz tasarlanması gerekmektedir. Bu amaçla çalışmada tasarlanan dil ve ırk tanıma sistemi PyQt5 üzerinden tasarlanan arayüz üzerinden kullanıma sunulmaktadır.

- **NumPy Veri Analiz Kütüphanesi**

NumPy kütüphanesi, Python programlama dilinde bilimsel hesaplama ve veri işleme için kullanılan açık kaynaklı bir kütüphanedir. "Numerical Python" kelimelerinin kısaltması olan NumPy, çok boyutlu dizilerin (n-arrays) yanı sıra bu diziler üzerinde yüksek performanslı matematiksel işlemler yapmak için bir dizi fonksiyon ve araç sunmaktadır. NumPy kütüphanesi çalışmada kullanılacak verilerin ön işleme aşamalarında kullanılmıştır.

- **Librosa Ses Azaliz Kütüphanesi**

Librosa kütüphanesi, Python programlama dilinde ses işleme ve müzik analizi için kullanılan bir kütüphanedir. Özellikle ses dosyaları üzerinde işlem yapmak ve ses verilerini analiz etmek için tasarlanmıştır. Librosa, çeşitli ses dosyası formatlarını (MP3, WAV, FLAC, WAV vb.) okuyabilen ve içeriğini işleyebilen işlevler sağlamaktadır. Librosa, ses verilerinden çeşitli özellikler çıkarmak için işlevler sağlamaktadır. Bu özellikler, mel-frekans cepstrum katsayıları (MFCC), spektral merkez frekansı, spektral enerji, ritim özellikleri gibi sesin farklı yönlerini temsil eden özelliklerdir. Ayrıca Librosa, ses verilerinden spektrogramlar oluşturmak için işlevler sağlamaktadır. Librosa kütüphanesi çalışmada kullanılacak verilerin ön işleme aşamalarında kullanılmıştır.

- **Pydub Ses İşleme Kütüphanesi**

Pydub kütüphanesi, Python'da ses dosyaları üzerinde işlem yapmak için kullanılan bir kütüphanedir. Ses dosyalarını okuma, yazma, kesme, birleştirme, ses seviyesini ayarlama, efektler uygulama gibi birçok işlemi gerçekleştirmek için tasarlanmıştır. Pydub, ses işleme süreçlerini kolaylaştıran basit bir arayüz sunmakta ve arka planda ffmpeg veya avconv gibi dış yazılımları kullanarak ses dosyalarını işlemektedir. Pydub kütüphanesi çalışmada kullanılacak verilerin ön işleme aşamalarında kullanılmıştır.

- **Matplotlib Veri Görselleştirme Kütüphanesi**

Matplotlib, Python dilinde veri görselleştirme işlemleri için sıklıkla kullanılan bir kütüphanedir, özellikle bilimsel ve mühendislik projelerinde veri analizi ve sonuçların görselleştirilmesi için büyük önem taşımaktadır. Çeşitli grafik türlerini destekleyen esnek yapısıyla, farklı veri setlerini etkili bir şekilde görselleştirmek için ideal bir

araçtır. Matplotlib kütüphanesi eğitim sonuçlarının görselleştirilmesi ve spektrogram çizimi için kullanılmıştır.

- **Google Colab**

Google Colab, Google tarafından sunulan ücretsiz bir bulut tabanlı hizmettir ve Python programlama dilini kullanarak kod yazma, çalıştırma ve paylaşma imkânı sunmaktadır. Colab, Jupyter Notebook'u temel almakta ve kullanıcıların tarayıcı üzerinden Python kodu yazmasına, çalıştırmasına ve sonuçları görsel olarak görüntülemesine olanak tanımaktadır. Çalışmada Google Colab hizmetinin kullanılması ve eğitimin Google Colab üzerinden gerçekleştirilmesinin çeşitli teknik ve pratik avantajları bulunmaktadır. İlk olarak Google Colab kullanıcılara güçlü GPU ve TPU kaynakları sunmaktadır. Bu sayede derin öğrenme hesaplamaları hızlı bir şekilde gerçekleştirilmiştir.

- **Google Drive**

Google Drive, Google tarafından sunulan çevrimiçi bir bulut depolama ve dosya paylaşım hizmetidir. Kolay kullanımı, geniş depolama alanı, dosya paylaşım özellikleri bulunmaktadır. Çalışmada yetersiz depolama alanı sorununun önüne geçmek ve verilerin depolanması amacıyla kullanılmıştır.

3.2. Uygulamanın Gerçekleştirilme Aşamaları

Bu bölümde dil tanıma ve ırk tanıma sistemlerinin birleştirilmesi sonucunda birleştirilen sistemin nasıl çalıştığına dair blok diyagramı verilmiştir. Birleştirilen sistemin arayüz üzerinden kullanımı ve bu arayüzün tasarımının nasıl gerçekleştirildiği bu bölümde açıklanmıştır. Ayrıca sistemi oluşturan kod bloklarının açıklamaları verilmiştir.

Arayüz Tasarım Aşamaları

İrk ve dil tanıma sistemlerinin birleştirilip arayüz üzerinden kullanılması amacıyla grafiksel kullanıcı arayüzü (GUI) tasarlanmıştır. GUI tasarımı için ilk olarak Qt

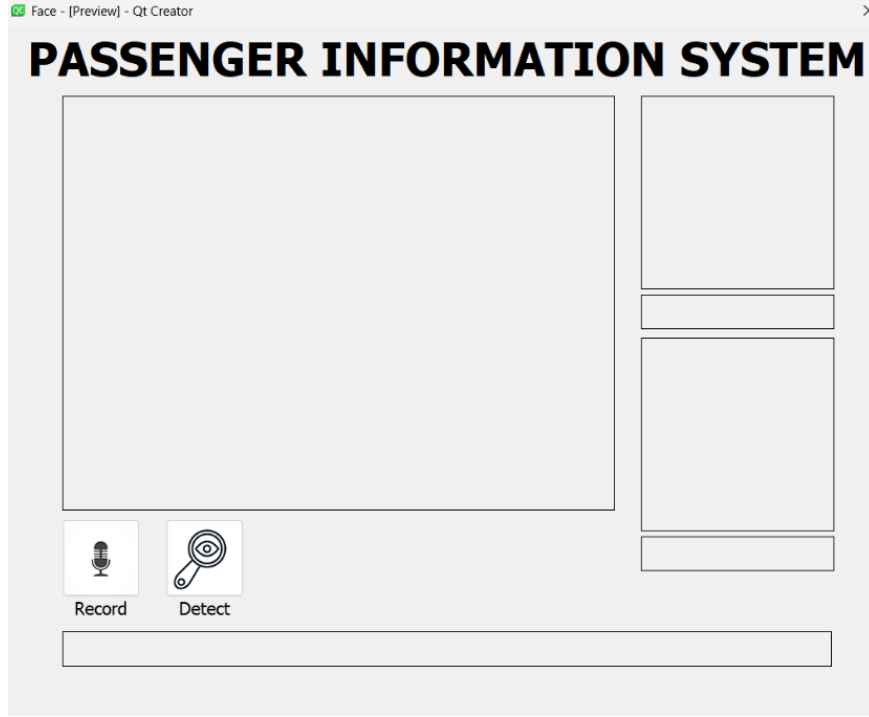
Creator uygulaması üzerinden arayüz görseli tasarlanmıştır. Tasarlanan arayüz Şekil 3. 2’de verilmiştir.

Qt Creator ile tasarlanan arayüz dosyası “uic.load.UiType” fonksiyonu ile Python sınıfına dönüştürülmüştür. “AudioRecordThread” sınıfı, arka planda ses kaydını gerçekleştiren bir iş parçacığını temsil etmektedir. “run” fonksiyonu ses kaydını başlatmış ve kayıt bittiğinde “finished” sinyalini yaymıştır. “record” fonksiyonu ise “sounddevice” kullanarak ses kaydetmiş ve WAV dosyasına yazmıştır.

“MainWindow” sınıfı ile kullanıcı arayüzü başlatılmış, eğitilen ırk ve dil tanıma modellerini yüklenmiş, kamera başlatılmış ve “QTimer” ile görüntü güncellenmiştir. Buton tıklamaları için sinyaller ve ses kaydı için bir arka plan iş parçacığı ayarlanmıştır.

Buton ikonlarının ayarlanması ve ikon boyutlarının düzenlenmesi “design” fonksiyonunda yapılmıştır.

Tespit fonksiyonu ile yüz ve ses tanıma işlemlerinin sonuçları kullanıcı arayüzüne yansıtılmıştır. Önce “detect_race” ile yüz, ardından “detect_language” ile ses tanıma yapılarak olasılıklar hesaplanmış ve hazırlanılan algoritma ile nihai tahmin belirlenmiştir.



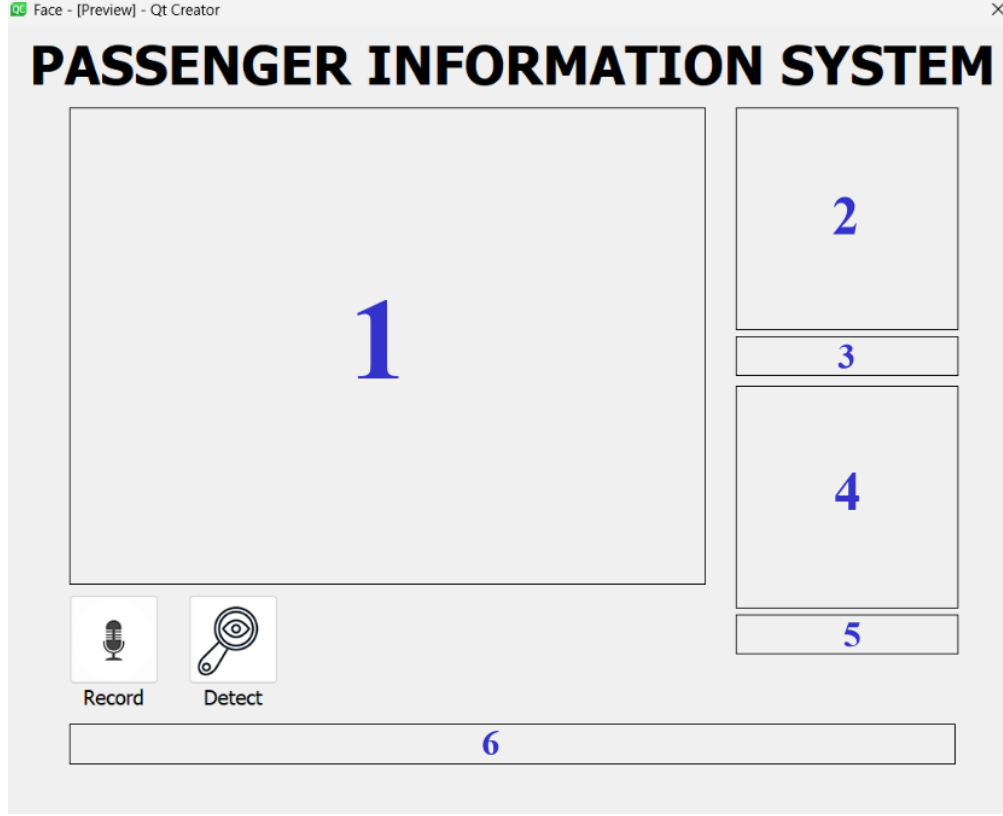
Şekil 3. 2. Tasarlanan GUI

Arayüz Kullanımı

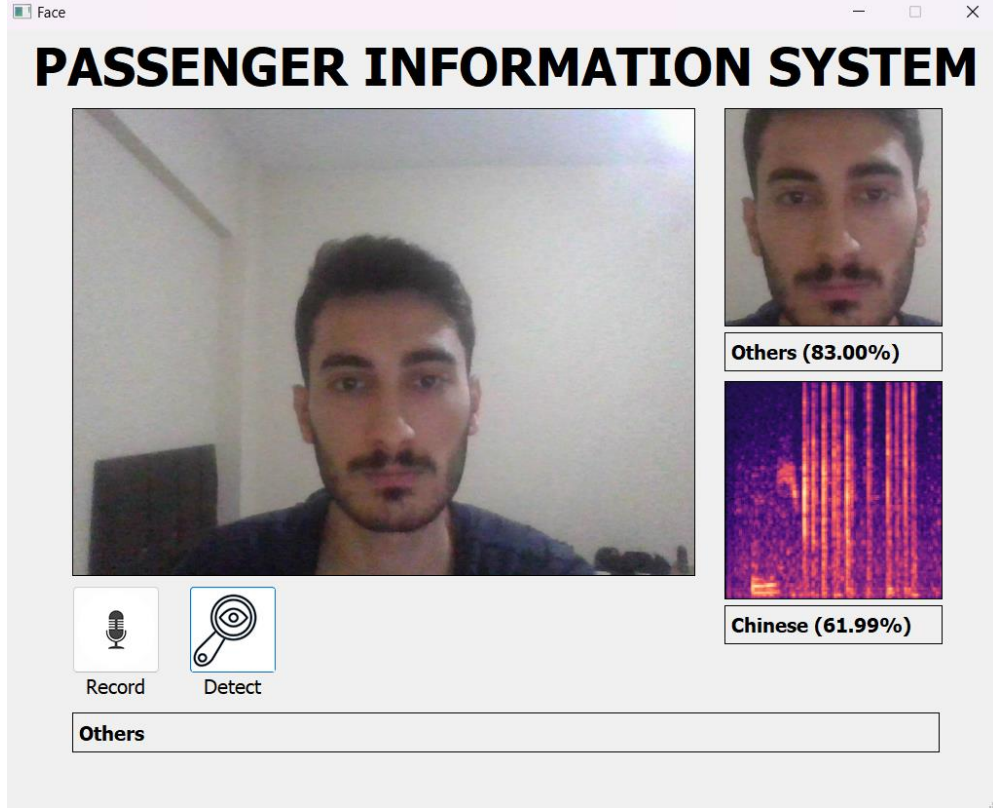
Çalışma kapsamında geliştirilen ırk ve dil tanıma sistemlerinin birleştirilip uygulamaya dökülmesi amacıyla bir arayüz tasarlanmıştır. Tasarlanan arayüz Şekil 3. 3'te gösterilmiştir. Şekil 3. 3'teki 6 numaralı alan genel bilgilendirme için ayrılmıştır. Ayrıca algoritmanın çıkışı yani görüntü ve ses destekli tahminin sonucu sağlanmıştır. Şekil 3. 3'teki 3 numaralı alan ırk, 5 numaralı alan ise dil tahmininin görüntülendiği alanı belirtmektedir. Şekil 3. 3'teki 1 numaralı alanda anlık kamera görüntüsü bulunmaktadır. Şekil 3. 3'teki 2 numaralı alanda yüz tespiti yapılan yüzün fotoğrafı bulunurken, 4 numaralı alanda dil tanıma işlemi yapılan ses klibinin spektrogram fotoğrafı yer almaktadır.

Arayüz programı ilk başlatıldığında kullanıcıdan "Record" butonuna basılması istenmiştir. "Record" butonuna basıldığında, 5 saniyelik bir ses kaydı alınmış ve spektrograma dönüştürülmüştür. Ardından "Detect" butonu aktif hale gelmiştir. Ses kaydı tamamlandıktan sonra kullanıcıdan "Detect" butonuna basması istenmiştir. "Detect" butonuna basıldığında kamerada yüz tespit edilirse, algoritma çalıştırılmış ve tahmin edilen dil sınıfı Şekil 3. 3'te 6 numaralı alana yazılmıştır. Yüz tespit edilmezse, "Yüz tespit edilemedi. Lütfen tekrar deneyin." şeklinde bir hata mesajı gönderilmiştir.

"Detect" butonuna basıldıktan 10 saniye sonra program kendini sıfırlamakta ve yeni bir kullanıcıyı beklemeye geçmektedir. Tasarlanan arayüze örnek görüntü ve ses girdisi için programın çalışması Şekil 3. 4'te gösterilmiştir.



Şekil 3. 3. Tasarlanan GUI alan bilgisi



Şekil 3. 4. Tasarlanan GUI örnek test bilgisi

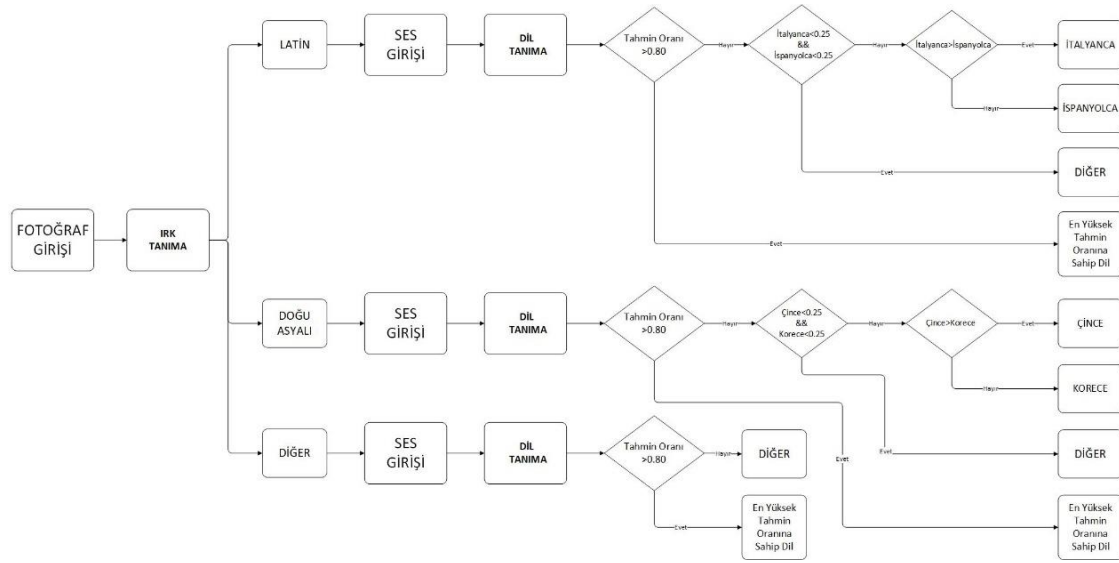
Birleştirilen Sistemin Çalışma Algoritması

Sistem, öncelikle fotoğraf girişini kullanarak kişi "Latin", "Doğu Asyalı" veya "Diğerleri" olarak sınıflandırılmıştır. Eğer kişi "Latin" olarak tanımlanmışsa, ses girişi alınmış ve dil tanıma aşamasına geçilmiştir. Dil tanıma modeline ses girdisi verilerek tahmin oranı hesaplanmıştır. Tahmin oranı %80'dan büyük olan dil, direkt olarak sonuç dili olarak belirlenmiştir. Eğer tahmin oranı %80'dan küçükse, ses girdisi diğer dillerle karşılaştırılmıştır. İtalyanca ve İspanyolca tahmin oranları %25'ten küçükse, sonuç "diğer" olarak belirlenmiştir. İtalyanca ve İspanyolca tahmin oranları %25'ten büyükse, iki dil arasında büyük olan dil sonuç dili olarak belirlenmiştir.

Eğer kişi "Asyalı" olarak tanımlanmışsa, ses girişi alınmış ve dil tanıma aşamasına geçilmiştir. Dil tanıma modeline ses girdisi verilerek tahmin oranı hesaplanmıştır. Tahmin oranı %80'dan büyük olan dil, direkt olarak sonuç dili olarak belirlenmiştir. Eğer tahmin oranı %80'dan küçükse, ses girdisi diğer dillerle karşılaştırılmıştır. Korece ve Çince tahmin oranları %25'ten küçükse, sonuç "diğer" olarak belirlenmiştir.

İtalyanca ve İspanyolca tahmin oranları %25'ten büyükse, iki dil arasında büyük olan dil sonuç dili olarak belirlenmiştir.

Eğer kişi "Diğer" olarak tanımlanmışsa, ses girişi alınıp dil tanıma aşamasına geçilmiştir. Dil tanıma modeline ses girdisi verilerek tahmin oranı hesaplanmıştır. Tahmin oranı %80'dan büyükse sistem direkt olarak sonuç dilini belirlemiştir. Eğer tahmin oranı %80'dan küçükse, en yüksek tahmin oranına sahip dil sonuç dili olarak belirlenmiştir. Algoritmanın blok diyagramı Şekil 3. 5'te gösterilmiştir.



Şekil 3. 5. Birleştirilen sistemin çalışma algoritması

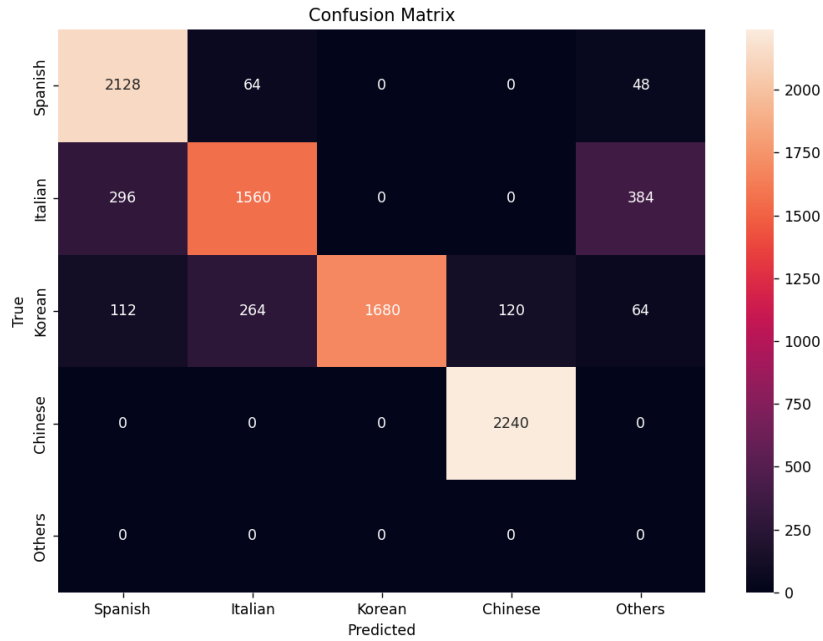
3.3. Uygulama Sonuçları ve Yorumlanması

Çalışmada, derin öğrenme modellerinden biri olan CNN modelini kullanan YOLOv8x-cls algoritması kullanarak ırk ve dil tanıma tabanlı yolcu bilgilendirme sistemi tasarlanmıştır. Irk ve dil tanıma sistemleri için ayrı ayrı ön işleme ve eğitim aşamaları gerçekleştirilmiştir. Bu iki sistemi bir arada kullanmak için grafiksel kullanıcı arayüzü (GUI) tasarlanmış ve bu arayüz üzerinden ırk ve dil tanıma sisteminin bir arada kullanımı sağlanmıştır. İki sistemin beraber kullanılması sonucunda tespit edilen dil, arayüz üzerinden görüntülenmektedir.

Arayüz üzerinden sistemi test etmek amacıyla test verileri hazırlanmıştır. Bu test veri seti, eğitim aşamasında ırk ve dil tanıma modellerine girdi verisi olarak verilmeyen

uygun veriler içermektedir. Irk tanıma için hazırlanan test veri setinde "Doğu Asyalı", "Latin" ve "Diğerleri" olmak üzere 3 sınıf bulunmaktadır. Ses tanıma için hazırlanan test veri setinde ise "Çince", "Korece", "İspanyolca" ve "İtalyanca" olmak üzere 4 sınıf bulunmaktadır.

Hazırlanan veri setleri ile yapılacak testler için bir test kodu geliştirilmiştir. Hazırlanan test kodu, her bir dil dosyasını ırk tanıma setindeki tüm fotoğraflarla eşleştirip test etmektedir. Yapılan testler sonucunda algoritma iyileştirilmiş ve en güncel algoritma ile son test gerçekleştirilmiştir. Son testin sonucunda Şekil 3. 6'da gösterilen konfüzyon matrisi elde edilmiştir. Elde edilen konfüzyon matrisinin doğruluk değeri 3.1 numaralı denklem kullanılarak hesaplanmıştır.



Şekil 3. 6. Birleştirilen sistemin test sonucunda elde edilen konfüzyon matrisi

$$\text{Accuracy} = \frac{2128+1560+1680+2240}{2240+2240+2240+2240} = 0,849 \quad (3.1)$$

Çalışmada geliştirilen ırk ve dil tanıma sistemleri yapılan testler sonucunda hedeflenen %80 doğruluk oranını başarıyla elde etmiştir. Irk ve dil tanıma sistemlerinin birleştirilmesi sonucunda sistemin genel başarısı ise %85 doğruluk oranına ulaşmıştır.

BÖLÜM 4. SONUÇLAR VE ÖNERİLER

Bu bölümünde tüm çalışmanın sonucunda elde edilen bulgular yorumlanarak çeşitli öneriler sunulmuştur. Çalışma sonuçlarının gerçekçi kısıtlar açısından analizi yapılmıştır.

4.1. Sonuçlar

Çalışmada, derin öğrenme modellerinden biri olan CNN modelini kullanan YOLOv8x-ccls algoritması kullanarak ırk ve dil tanıma tabanlı, tasarlanan GUI üzerinden yolcunun konuştuğu dili tespit edip bu dile uygun olarak bilgi veren sistem tasarımı yapılmıştır. Sistemde kullanıcının ırk ve dil tanımlaması sonucu çifte doğruluk yaparak nihai karar olarak konuşulan dil tanımlanması yapılmaktadır. Yolcu bilgilendirme metni arayüze dahil edilmemiştir. Çünkü sistemin kullanıldığı alana göre bilgilendirme metni değişiklik göstermektedir. Sistem ırk ve dil tanıma yaparak çifte doğruluk ile konuşulan dili tespit etmektedir. Sonuç hedeflenen doğruluk oranına ulaşmıştır.

Çalışma 4 dil sınıfı üzerinden yapılmıştır. Bunun sebebi istenilen uygunlukta veri seti bulunamaması ve yeterli donanıma sahip olunmamasıdır.

İrk tanıma sisteminde girdi fotoğrafının 10 lux'ten düşük olduğu durumlarda sistem performansı etkilenmektedir.

Dil tanıma sisteminde girdi sesinin kötü bir mikrofon üzerinden kayıt alınması veya kullanıcının gürültülü bir ortamda olması sistem performansını etkilemektedir.

4.2. Öneriler

Çalışmada derin öğrenme yöntemi kullanarak ırk ve dil tanıma tabanlı kullanıcının konuştuğu dile uygun olarak bilgilendirme yapan yolcu bilgilendirme sistemi tasarımı yapılmıştır. Çalışmada incelenmemiş ancak potansiyel olarak önemli olabilecek bazı konular ve yöntem önerileri şunlardır.

1. Farklı Derin Öğrenme Modelleri: Çalışmada derin öğrenme modeli olarak CNN kullanılmış ve çalışmalar CNN modeline göre yapılmıştır. Farklı derin öğrenme modelleri üzerinde çalışmalar yapılabilir.
2. Makine Öğrenmesi: Irk ve dil tanıma modeli derin öğrenme yöntemi ile eğitilmiştir. Çalışmada makine öğrenmesi yöntemleri kullanılmamıştır. Makine öğrenmesi yöntemleri incelenerek eğitim yapılabilir.
3. Farklı Ses İşleme Yöntemleri: Çalışmada ses verilerinden özellik çıkarımı yapmak için ses klipleri spektrogramlara dönüştürülmüştür. Ses verilerinin farklı işleme yöntemleri kullanılarak ses analizi yapılabilir.

4.3. Sonuçların Sağlık, Çevre ve Güvenlik Açısından Analiz

Çalışmanın belirlenen sağlık, çevre ve güvenlik kısıtları sistemin eğitim, geliştirme ve test aşamalarında dikkatle incelenmiş ve çalışmanın yazılım projesi olmasından dolayı herhangi bir sorun tespit edilememiştir. Çalışmanın ilerleyen zamanlarda geliştirilerek yolcu terminallerine veya bunun gibi kamusal alanlarda kullanılması halinde sağlık, çevre ve güvenlik açısından herhangi bir sorun teşkil edeceği düşünülmemektedir. Irk tanıma sistemi geliştirilip yabancı uyruklu suçlu tespiti için güvenlik amacıyla kullanılabilir.

BÖLÜM 5. KAYNAKLAR

- [1] Belcar, David, Petra Grd, and Igor Tomičić. "Automatic ethnicity classification from middle part of the face using convolutional neural networks." *Informatics*. Vol. 9. No. 1. MDPI, 2022.
- [2] Abdulwahid, Abdulwahid Al. "Classification of Ethnicity Using Efficient CNN Models on MORPH and FERET Datasets Based on Face Biometrics." *Applied Sciences* 13.12 (2023): 7288.
- [3] Hadžić, Ibrahim. "RECOGNITION OF NATURAL LANGUAGE TYPE USING TENSORFLOW."
- [4] Singh, Gundeeep, et al. "Spoken language identification using deep learning." *Computational Intelligence and Neuroscience* 2021 (2021).
- [5] Karkkainen, Kimmo, and Jungseock Joo. "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation." *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021.
- [6] Valk, Jörgen, and Tanel Alumäe. "VoxLingua107: a dataset for spoken language recognition." *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021.
- [7] Satyanarayana, G. V., et al. "Spoken Language Detection Using Deep Learning.
- [8] Nema, B. M., & Abdul-Kareem, A. A. (2018). Preprocessing signal for speech emotion recognition. *Al-Mustansiriyah Journal of Science*, 28(3), 157-165.
- [9] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20), 4396.

- [10] Gupta, J., Pathak, S., & Kumar, G. (2022, May). Deep learning (CNN) and transfer learning: A review. In *Journal of Physics: Conference Series* (Vol. 2273, No. 1, p. 012029). IOP Publishing.
- [11] Hu, X., Men, R., Li, G., & Jin, Z. (2019, July). Deep-autocoder: Learning to complete code precisely with induced code tokens. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 159-168). IEEE.
- [12] DOĞAN, F., & TÜRKOĞLU, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- [13] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE access*, 7, 53040-53065.
- [14] Li, G., Hari, S. K. S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J., & Keckler, S. W. (2017, November). Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-12).
- [15] O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- [16] Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 92-108.
- [17] Sun, M., Song, Z., Jiang, X., Pan, J., & Pang, Y. (2017). Learning pooling for convolutional neural network. *Neurocomputing*, 224, 96-104.

- [18] Garbin, C., Zhu, X., & Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia tools and applications*, 79(19), 12777-12815.
- [19] Alabassy, B., Safar, M., & El-Kharashi, M. W. (2020, April). A high-accuracy implementation for softmax layer in deep neural networks. In *2020 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)* (pp. 1-6). IEEE.
- [20] Samir, S., Emary, E., El-Sayed, K., & Onsi, H. (2020). Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries. *Information*, 11(5), 275.
- [21] Alzamily, J. I., Ariffin, S. B., & Abu-Naser, S. S. (2022). Classification of Encrypted Images Using Deep Learning–Resnet50. *Journal of Theoretical and Applied Information Technology*, 100(21), 6610-6620.
- [22] Younis, A., Qiang, L., Nyatega, C. O., Adamu, M. J., & Kawuwa, H. B. (2022). Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches. *Applied Sciences*, 12(14), 7282.
- [23] Jaworek-Korjakowska, J., Kleczek, P., & Gorgon, M. (2019). Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0-0).
- [24] Hasan, M. M., Ali, H., Hossain, M. F., & Abujar, S. (2020, July). Preprocessing of continuous bengali speech for feature extraction. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-4). IEEE.
- [25] Labied, M., Belangour, A., Banane, M., & Erraissi, A. (2022, March). An overview of automatic speech recognition preprocessing techniques. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 804-809). IEEE.

- [26] Smith, S. W. (1997). The scientist and engineer's guide to digital signal processing.
- [27] Parizad, A., & Hatziaioniu, C. (2021). Deep learning algorithms and parallel distributed computing techniques for high-resolution load forecasting applying hyperparameter optimization. *IEEE Systems Journal*, 16(3), 3758-3769.
- [28] Kumari, S., Gautam, A., Basak, S., & Saxena, N. (2023, December). YOLOv8 Based Deep Learning Method for Potholes Detection. In 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI) (pp. 1-6). IEEE.
- [29] Karkkainen, K., & Joo, J. (2021). Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1548-1558).
- [30] Ultralytics (2024). Ultralytics YOLO Docs <https://docs.ultralytics.com/models/yolov8/>
- [31] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*. CVPR 2001, vol. 1. IEEE, 2001, pp. I-I.
- [32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [33] Twine (November 21, 2023). Top 10 Open Source Facial Recognition libraries and tools <https://www.twine.net/blog/top-10-open-source-facial-recognition-library-and-tools/>

- [34] Ultralytics (2024). Augmentation Settings and Hyperparameters.
<https://docs.ultralytics.com/modes/train/#augmentation-settings-and-hyperparameters>

BÖLÜM 6. ÖZGEÇMİŞ

İbrahim Halil Özçakır, 03.02.2001 tarihinde Bolu’da doğdu. Lise eğitimini 2019 yılında Düzce Osman Kuyumcu Anadolu Lisesi’nde tamamladı. Üniversite eğitimine Sakarya Üniversitesi Elektrik Elektronik Mühendisliği Bölümü’nde devam etmektedir. Aynı zamanda Bilişim Sistemleri Mühendisliği bölümünden çift anadal yapmaktadır. Anadal lisans eğitimi 3. sınıfta Teknofest Tarımsal İnsansız Kara Aracı yarışmasına katılmış ve projede elektronik sistemler, güç dağıtım kartı baskı devre çizimi çalışmalarında bulunmuştur. Lisans 4. sınıf bahar dönemi boyunca Arifiye/Sakarya konumunda bulunan BMC Otomotiv San. ve Tic. Firmasında uzun dönem staj yapmıştır. Staj esnasında Fırtına Obüs’ü mermi doldurma sistemi arayüzü ve simülasyon projesini yapmıştır.

Burak Okur, 11.03.2001 tarihinde İstanbul Üsküdar ilçesinde doğdu. Ortaokul ve lise eğitimini Samsun'un Bafra ilçesinde tamamladı. Üniversite eğitimini Sakarya Üniversitesi Elektrik Elektronik Mühendisliği Bölümü'nde tamamladı. IEEE SAÜ Helios Rover Team ile 2022 yılında Teknofest Tarımsal İKA yarışmasına katıldı ve yarışmayı takım olarak 4.lük ile bitirdi. Zorunlu stajlarını HAVELSAN ve ASELSAN firmalarında yaptı. Lisans 4. sınıfta, 6 ay boyunca Kahramankazan/Ankara konumunda bulunan TUSAŞ firmasında aday mühendislik yaptı. Bu süre zarfında Rüzgâr Tüneli ve Gökbeş Helikopter projelerinde çalıştı. Teknolojiye meraklı, yeniliklere ve öğrenmeye açık biridir.

BÖLÜM 7. EKLER

EK A. Deney Tasarımı Açıklamaları

Çalışmada yapılan sistem bilgisayar üzerinden çalıştırılan arayüz olduğu için deney kısmı, arayüzü farklı parametreler üzerinden test etmek üzerine kurgulanmıştır. Test için ilk olarak arayüzün çalışma hızı test edilmiştir. Ayrıca bozuk veriler üzerinden sistemi test etmek amacıyla dil tanıma sistemi için farklı dillerde veriler ve bozuk mikrofondan veya gürültülü ortamda kaydedilen sesler kullanılmıştır. Irk tanıma sistemi için girdi görüntüleri farklı açılarda ve parlaklık oranında değişiklikler yapılarak sistem test edilmiştir. Test sonucunda elde edilen veriler analiz edilmiş ve sistemin yüksek performanslı çalışabilmesi için verilerin uygunluğu belirlenmiştir. Sistemin hazırlanan normal veri seti ile performans testi 3.3. Uygulama Sonuçları ve Yorumlanması” kısmında yapılmıştır.

EK B. IEEE Etik Kurallar Onay Formu



IEEE üyeleri olarak bizler bütün dünya üzerinde teknolojilerimizin hayat standartlarını etkilemesindeki önemin farkındayız. Mesleğimize karşı şahsi sorumluluğumuzu kabul ederek, hizmet ettiğimiz toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmayı söz verdiğimizizi ve aşağıdaki etik kuralları kabul ettiğimizi ifade ederiz.

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

Bu çalışma IEEE Etik ilkelerine bağlı kalınarak hazırlanmıştır.

Ad Soyad	Öğrenci No	İmza
İbrahim Halil Özçakır	B190100015	
Burak Okur	B190100042	

EK C. Çalışmanın Ulusal veya Uluslararası Standartlar İlişkisi

EK E. Çalışma Yönetimi

Bu bölümde, iş-zaman çizelgesi ve/veya iş-görev paylaşımı, başarı ölçütleri ve risk yönetimi belirtilmektedir.

İş Zaman Çizelgesi ve/veya İş Görev Paylaşımı

Çalışmaya ait iş zaman çizelgesi ve görev paylaşımı Tablo 7. 1’de verilmiştir.

Tablo 7. 1. İş zaman çizelgesi tablosu

İP No	İş Paketlerinin Adı ve Hedefleri	Kim(ler) Tarafından Gerçekleştirileceği	Zaman (Hafta.)	Başarı Ölçütü ve Projenin Başarısına Katkısı
1	İrk görsellerinin ve farklı dillerdeki ses verilerinin bilgisayar ortamında toplanması	İbrahim Halil ÖZÇAKIR Burak OKUR	1 hafta	Verilerin internetten temin edilmesi (%10)
2	Görsel verilerin ön işleme	İbrahim Halil ÖZÇAKIR Burak OKUR	3 hafta	Yüzün orta kısmının alınması (%15)
3	Görsel verilerden derin öğrenme mimarisiyle özellik çıkarma ve sınıflandırma	İbrahim Halil ÖZÇAKIR Burak OKUR	3 hafta	4 farklı ırka ait özelliklerinin derin öğrenme tekniğiyle tespiti ve sınıflandırılması (%20)
4	Ses verilerinin ön işleme	İbrahim Halil ÖZÇAKIR Burak OKUR	3 hafta	Ses verilerinin 5 sn’lik kayıtlar şeklinde kırılması ve spektrogram Dönüştürülmesi (%20)
5	Ses verilerinden derin öğrenme mimarisiyle özellik çıkarma ve sınıflandırma	İbrahim Halil ÖZÇAKIR Burak OKUR	3 hafta	4 farklı dile ait özelliklerin derin öğrenme tekniğiyle tespiti ve sınıflandırılması (%20)
6	Arayüz tasarımı sonrasında sistemlerin birleştirilmesi ve test	İbrahim Halil ÖZÇAKIR Burak OKUR	3 hafta	Arayüz tasarımı ve sonuçların test edilmesi (%15)

Başarı Ölçütleri

Aşağıda, derin öğrenme çalışmasının başarı ölçütleri ve önem dereceleri birlikte verilmiştir.

1. Veri hazırlığı: Çalışmada kullanılan veri setinin hedef probleme uygun olması, yeterli sayıda veri içermesi, verilerin tutarlı ve etiketli olması (Başarıya katkısı: % 40)
2. Model geliştirme: Hedef probleme uygun model mimarisinin seçimi, hiper parametrelerin optimize edilmesi, aşırı uyumu engellemek için düzeltme tekniklerinin uygulanması (Başarıya katkısı: % 20)
3. Eğitim süresi: Eğitim süresince kayıp ve doğruluk fonksiyon metriklerinin incelenmesi, modelin yeterli süre eğitilmesi (Başarıya katkısı: % 15)
4. Değerlendirme: Modelin doğrulama ve test setlerinde yüksek performans göstermesi. (Başarıya katkısı: % 15)
5. Uygulama: Modelin gerçek dünyada test edilip beklenen performansı göstermesi, modelin hızlı çalışması ve büyük veriler ile başa çıkabilmesi (Başarıya katkısı: % 10)

Risk Yönetimi

Çalışmada oluşturulan CNN mimarisi beklenen performansın sağlanamaması durumunda B planı olarak farklı mimarilerin kullanılması planlanmaktadır.

Çalışma sonunda ortaya çıkan tüm sisteme ait doğruluğun hedeflenenden düşük olması durumunda B planı olarak veri tabanının genişletilmesi ve veri çoğaltma işlemleri uygulanacaktır.

Tablo 7. 2. Risk yönetim tablosu

İP No	En Önemli Riskler	Risk Yönetimi (B Planı)
1	Oluşturulan CNN mimarilerinin beklenen performansı sağlamaması	Farklı mimarilerin kullanımı
2	Tüm sisteme ait doğruluğun hedeflenenden düşük olması	Veri tabanının genişletilmesi

Çalışmanın Sürdürülebilir Kalkınma Açısından Önemi

Yolcu terminallerinde (Havalimanı, Otogar, Tren Garı) yolcuların konforunu ve memnuniyetini arttırmak amacıyla tasarlanacak bu sistem, yabancı bir ülkede yolculuk yapan insanların dil bariyerleri nedeniyle bilgilendirme alamamasının önüne geçmekte ve yolculuk deneyimlerinin olumlu yönde artmasını hedeflemektedir.

Maliyet Değerlendirmesi

Tablo 7. 3. Maliyet değerlendirme tablosu

Maliyet Türü	Maliyet Miktarı (TL)	Maliyet Gerekçesi
Sarf Malzeme	6500	Depolama alanının yetersizliği ve bilgisayarın işlem hacmini arttırmak için Ram ve SSD bellek alımı. Isınma sorununa karşı laptop soğutucusu alımı.
Makina/Teçhizat (Demirbaş)		
Hizmet Alımı	2000	Eğitim için gerekli GPU desteği sağlamak amacıyla Google Colab alımı. Depolama desteği için Google Drive alımı
Ulaşım		
TOPLAM		

BİTİRME ÇALIŞMASI SON KONTROL FORMU

		EVET	HAYIR
1	Beyan formu çalışmada bulunuyor mu?	<input type="checkbox"/>	<input type="checkbox"/>
2	Özet bölümü en az 150 kelime içermekte mi?	<input type="checkbox"/>	<input type="checkbox"/>
3	Simge ve Kısaltmalar uygun şekilde yazılmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
4	Şekiller listesi formata uygun şekilde hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
5	Tablolar listesi formata uygun şekilde hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
6	İş paketleri ve zaman çizelgesi çalışmada verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
7	Başarı ölçütlerinden bahsedilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
8	Risk yönetimi ve planlaması bulunmakta mı?	<input type="checkbox"/>	<input type="checkbox"/>
9	Sonuçların sağlık, çevre ve güvenlik açısından analizi gerçekleştirilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
10	Deney Tasarımı Açıklamaları verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
11	IEEE Etik kurallar onay formu imzalanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
12	Çalışmanın Ulusal ya da Uluslararası standartlar ilişkisi verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
13	Seminer, Kongre, Konferans veya Teknik Gezi katılım kanıtı mevcut mu?	<input type="checkbox"/>	<input type="checkbox"/>
14	Kaynaklar bölümü formata uygun hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
15	Yazım kılavuzuna aykırı durumlar mevcut mu?	<input type="checkbox"/>	<input type="checkbox"/>

Bu çalışma tarafımdan incelenmiş olup kontrol sonuçları yukarıda verildiği gibidir.

İnceleyen :

Tarih :

İmza :