T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

LİSANS BİTİRME PROJESİ

OPTİK KARAKTER TANIMA SİSTEMİ

Şule MEŞE

Bilgisayar Mühendisliği Bölümü

HAZİRAN 2024

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

LİSANS BİTİRME PROJESİ

OPTİK KARAKTER TANIMA SİSTEMİ

Şule MEŞE

DANIŞMAN
Doçent Doktor İlyas ÖZER

Bilgisayar Mühendisliği Bölümü

HAZİRAN 2024

ONAY

Şule Meşe tarafından hazırlanan "**Optik Karakter Tanıma Sistemi**" adlı proje çalışması …/…/… tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümünde LİSANS BİTİRME PROJESİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Unvanı, Adı ve Soyadı (Danışman)	_____
Unvanı, Adı ve Soyadı (Üye)	_____
Unvanı, Adı ve Soyadı (Üye)	_____
Unvanı, Adı ve Soyadı (Üye) (varsa)	_____
Unvanı, Adı ve Soyadı (Üye) (varsa)	_____

Doç. Dr. Rafet DURGUT

Bölüm Başkanı

ÖZET

Teknolojik gelişmeler yaşamın her alanında dijitalleşmeyi zorunlu kılmıştır. Dijitalleşmenin etkisiyle gündelik hayatta ve iş hayatında yapılan işlemler daha hızlı ve verimli gerçekleştirilmektedir. Bu dönüşümün kritik bir bileşeni olarak iş süreçlerinde ortaya çıkan belgelerin dijital ortamda saklanması verilerin yönetim ve erişimi konusunda kolaylık sağlamaktadır. Bu bağlamda Optik Karakter Tanıma (Optical Character Recognition – OCR) teknolojisi basılı metin görüntülerinin dijital formata dönüştürülmesinde kritik bir rol oynamaktadır. OCR teknolojisi basılı metinlerin dijital hale getirilmesi, düzenleme ve arama yapılabilir bir formata dönüştürülmesini sağlamaktadır. Bu çalışmada ilk olarak OCR teknolojisi için kullanılan ML Kit Text Recognition API ile gerçek zamanlı bir OCR sistemi geliştirilmiştir. Son olarak basılı metin görüntülerinden metinsel verileri tespit ederek optik karakterleri tanıyan bir OCR sistemi geliştirilmiştir. Bu doğrultuda elde edilen görüntülerin ön işlenmesi, görüntü üzerinde segmentasyon yapılarak metin görüntüsünden satırların ve kelimelerin sınırlarının belirlenmesi, yinelemeli sinir ağları ile metin tanıma işlemi, ve elde edilen tahmin sonuçlarının sıralanarak metnin elde edilmesi basamakları örnek bir görüntü üzerinde gerçekleştirilmiştir.

TEŞEKKÜR

Bu tez çalışmasının gerçekleştirilmesinde, değerli bilgi ve rehberlikleri ile bana yol gösteren saygıdeğer danışman hocam Doçent Doktor İlyas Özer'e en derin teşekkürlerimi sunarım. Hocamın göstermiş olduğu sabır, anlayış ve akademik destek, bu çalışmanın tamamlanmasında büyük rol oynamıştır. Araştırmam boyunca bana gerekli olanak ve malzemeleri sağlayan üniversitemiz yönetimine ve tüm akademik personele teşekkür ederim.

ŞULE MEŞE

İÇİNDEKİLER

Sayfa

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR LİSTESİ

A2iA: Artificial Intelligence Analysis

APK: Android Paket Kiti

API: Uygulama Programlama Arayüzü

AAB: Android Uygulama Paketi

BLSTM: Çift Yönlü LSTM (Bidirectional LSTM)

CNN: Evrişimli Sinir Ağı (Convolutional Neural Network)

CER: Karakter Hatası Oranı (Character Error Rate)

CSV: Virgülle Ayrılmış Değerler (Comma-Separated Values)

CTC: Bağlantılı Zamanlı Sınıflandırma (Connectionist Temporal Classification)

GAN: Üretici Karşıtlığı Ağı (Generative Adversarial Network)

Gradle: Birleştirme Aracı

IDE: Entegre Geliştirme Ortamı

JPEG: Birleştirilmiş Fotoğraf Uzmanları Grubu

LSTM: Uzun Kısa Süreli Bellek (Long Short-Term Memory)

Matplotlib: Python Görselleştirme Kitaplığı

minSdkVersion: Minimum Yazılım Geliştirme Kiti Sürümü

ML: Makine Öğrenmesi

ML Kit: Machine Learning Kit

Numpy: Sayısal Python

OCR: Optik Karakter Tanıma

OpenCV: Açık Bilgisayarlı Görü

OpenJDK: Açık Kaynak Java Geliştirme Kiti

ONNX: Açık Yapay Sinir Ağı Değişim Biçimi (Open Neural Network Exchange)

OpenJDK: Açık Kaynak Java Geliştirme Kiti

Otsu: Otsu eşikleme yöntemi

PDF: Portable Document Format

PNG: Taşınabilir Ağ Grafikleri

RGB: Red, Green, Blue

SDK: Yazılım Geliştirme Kiti

SLP: Tek Katmanlı Algılayıcı (Single-Layer Perceptron)

Streamlit: Birleşik Python Web Çerçevesi

TensorFlow: Google'ın Açık Kaynak Makine Öğrenimi Kitaplığı

UI: Kullanıcı Arayüzü

XML: Genişletilebilir Biçimlendirme Dili

1. GİRİŞ

Bu çalışma OCR konusu üzerine odaklanmaktadır. OCR Optik Karakter Tanıma teriminin kısaltmasıdır. OCR Optik Karakter Tanıma basılı veya el yazısı metin içeren dökümanlar, pdf dosyaları, dijital görüntüler gibi farklı belge türlerini düzenlenebilir ve aranabilir bir dijital formata dönüştüren teknolojidir [1]. OCR teknolojisi dijital görüntüden karakter tanıma işlemi gerçekleştirir [1]. Bu durumun sonucunda fiziksel belgenin metinsel içeriği dijital ortamda saklanması, metin üzerinde düzenleme yapılması veya metin içinde verilerin aranması mümkün olmaktadır [1]. OCR teknolojisi geniş kullanım alanına sahiptir. Bu alanlara plaka tanıma sistemleri, el yazısı tanıma, captcha aşma uygulamaları, doğal sahne görüntülerinin tanınması gibi örnekler verilebilir. Bilgisayar donanımları geliştikçe işlem hızları artmaktadır. Kamera, tarayıcı gibi ekipmanların gelişmesi kaliteli görüntüler elde etme konusunda başarıyı arttırmaktadır. Bu gibi durumların yansıması olarak, Optik Karakter Tanıma (OCR) konusu ilgi çekici olmaktadır [2]. Bu bağlamda çalışmanın konusu basılı metinlerden metinsel ifadeleri tanıyan bir OCR sisteminin geliştirilmesine odaklanmaktadır.

Çalışmanın amacı OCR sisteminin geliştirilmesi olacaktır. OCR işlemi metin içeren bir çok belge türünü kapsadığı için çalışmanın amacı spesifik olarak kitap metni gibi arka planında öğe içermeyen belge türlerinin metinsel verilerinin tanınmasına yönelik olacaktır. Bir bilgisayar sisteminin ona verilen görüntü hakkındaki bilgisi dijital görüntünün piksel değerleri ile sınırlıdır. Görüntünün metinsel içeriğini çıkartmak için görüntünün işlenerek karakterlerinin tanınması gerekmektedir. Bu noktada OCR teknolojisi kritik öneme sahiptir. OCR sisteminin görüntüden metin tanıma işlemini gerçekleştirmesi için API ve makine öğrenmesinin alt dalı olan derin öğrenme yöntemlerinden yararlanılarak metin tanıma yapılması amaçlanmaktadır. OCR uygulamalarında belgenin çekim açısı , farklı yazı stilleri içermesi, görüntüde bulanıklık vb. durumlar sebebiyle %100 doğruluk oranına ulaşmak zor

olmaktadır.[2] Bu nedenle çalışmada kabul edilebilir bir doğruluk elde edilmesi amaçlanmaktadır.

Ele alınan problemin tanıtımı ve OCR'ın önemi noktasında OCR sisteminin ortaya çıkmasında payı olan belli başlı faktörler vardır. Teknolojik gelişmeler dünyaya yön vererek dijitalleşmeyi zorunlu kılmıştır. Fiziksel belgeler depolanmak için fiziksel ortama gereksinim duymaları, içinden bir veri aramanın zaman açısından maliyeti, yönetiminin zor olması , bilgisayar ortamına el ile geçirilmesinin yarattığı zorluklar, erişilebilirliğinin kısıtlı olması gibi faktörlerle birlikte ocr kullanarak düzenlenebilir metin verisi içeren dijital formatta saklanmaya başlamıştır. Bu durum belgelerin hem depolama hem yönetimini kolaylaştırmaktadır [6].

Ele alınan problemler OCR performansı noktasında aşağıdaki gibi özetlenebilir. Karakter tanıma bilgisayar sistemleri için karmaşık bir görevdir. Çünkü karakter sayısı kadar sınıf vardır. Karakterler arasındaki benzerlikler karakter sınıflandırmada hataya yol açmaktadır. Örnek olarak "O" harfi ve "0" rakamı, "I" harfi ve "1" rakamı ve "B" harfi "8" rakamının benzerlikleri sınıflandırmayı zorlaştırmaktadır [3]. Bunun dışında OCR gerçekleştirilecek olan belgenin net, yüksek çözünürlükte ve doğru açıda görüntüsünün elde edilmesi için çekim açısının ve çekim ortamının ayarlanması önem taşır [4]. Görüntü sınırlarında meydana gelen bozulmalar diğer bir deyişle gürültüler ve normal olmayan ışık şiddeti metin tanıma performansını etkileyen yaygın problemlerdir [5].

Çalışmada kullanılacak yöntemler noktasında, Optik Karakter Tanıma Sistemi çalışmasında yaklaşım olarak öncelikle OCR uygulamalarında yaygın kullanılan Google Text Recognition ML Kit API ile OCR sistemi geliştirilmesi planlanmaktadır. İkinci aşamada OCR sistemlerinde yüksek doğruluk ve hızı ile ön plana çıkan makine öğrenimi yöntemi olan derin öğrenme tabanlı OCR sistemi geliştirilmesi planlanmaktadır.

İlk yaklaşım ele alınırsa Google ML Kit görüntü ve videodan metin tanıma gerçekleştiren API'dir. Latin alfabesi , Çince, Japonca gibi çeşitli dilleri destekler. Metin tanıma gerçekleştirmesi yanında metnin dilini de tanımlar. Çevrimiçi ve çevrimdışı metin tanıma hizmeti sağlar. Metin tanıma işlemi için metni öğe, satır,

paragraf gibi bloklara ayırarak segmentasyon gerçekleştirir. Sonrasında bu blokların her birinin sınırlayıcı kutularının çizilmesi, koordinatlarının tespit edilmesi, metninin tanınması ve hangi dilde olduğunun tespitini gerçekleştirir . Ayrıca Android ve iOS cihazlarda veya web uygulamaları için kullanılabilmektedir [7].

Bir diğer yaklaşımda yapay sinir ağı mimarisi kullanılarak OCR sistemi geliştirilmesi üzerinde durulmuştur. Yapay sinir ağı insanlara özgü olan düşünme, gözlem yapma, veriler arasında ilişki kurma gibi nitelikleri gerçekleştirmek amacıyla insan biyolojik sisteminin çalışma yapısının taklit edilerek bilgisayar sisteminde modellenmesidir. Yapay sinir ağı elde ettiği bilgileri kullanarak genelleme yapar. Bu sayede daha önce görmediği verilerle karşılaştığında deneyimlerinden faydalanarak tahmin gerçekleştirebilir [5,8]. Bu noktada çok katmanlı yapay sinir ağı anlamına gelen derin öğrenme mimarisi kullanılmıştır [5].

CNN "Convolutional Neural Network" teriminin kısaltmasıdır. Derin öğrenme alanında kullanılır. Özel bir yapay sinir ağı türüdür. Görüntülerden özellikler çıkartmakta başarılıdır [9]. Genellikle metin tanıma, ses tanıma, yüz tanıma, nesne tanıma gibi görüntü işleme görevlerinde yaygın olarak kullanılmaktadır.

RNN "Recurrant Neural Network" teriminin kısaltmasıdır. RNN sıralı verilerdeki örüntüleri anlamak ve zaman serisi problemleriyle başa çıkmak üzere tasarlanmıştır. Geleneksel yapay sinir ağlarında girdi ve çıktılar arasında ilişki yoktur. Yani tüm girdiler ve tüm çıktılar bağımsızdır. Ancak RNN girdiler arasında ilişki kurar ve hafızaya sahiptir. Bu nedenle geçmişte elde ettiği bilgileri hatırlayarak aşamaları takip eder ve her aşamada elde ettiği bilgileri hafızaya kaydeder. Bu şekilde zamansal bağımlılıkları işleme konusunda başarılıdır. Bu sayede zaman serisi, dil modeli analizi konularında yaygın olarak kullanılır. RNN mimarileri uzun vadeli bağımlılıkları ele almak noktasında yetersiz kaldığı için RNN'in daha başarılı versiyonu olan LSTM (Long Short Term Memory) ortaya çıkmıştır [10].

CNN'in görüntülerden özellikler çıkarma yeteneği ile RNN'in zaman serisi gibi sıralı verileri işleyebilme yeteneği birleştirilerek hibrit modeller oluşturulmaktadır. Bu sayede uzaysal ve zamansal veriler işlenebilir. Çalışmada da bu amaçla hibrit model kullanılmıştır. Girdi olarak kelime görüntüleri verilir. CNN katmanları görüntüden

karakterlerin dokusu, kenarı, köşesi gibi önemli özelliklerini çıkartır. Bu özellikler karakter tanıma açısından kritik rol oynar. RNN katmanları özellik haritasını kullanarak zamansal bağımlılıkları öğrenir ve karakter dizisinin anlamlı şekilde sıralanmasını sağlar. Her zaman aralığında her karakter için olasılık dağılımına göre en uygun sınıf tahmin edilir.

Gorski ve arkadaşları çek temizleme sürecindeki çeşitli ödeme belgelerini %65-85 arası doğrulukla tanıyan ticari banka çek tanıma sistemini ortaya atmıştır. Sistem çek, makbuz, havale çeki ve benzeri belge görüntülerinde çalışmaktadır. A2iA CheckReaderTM Belgenin stili , türü, el veya makine yazısı olması, rakamların sayı veya metin ile ifade edilmesi farketmeksizin çalışan bir sistemdir [11].

C. M. Ng ve arkadaşları Braille alfabesi ile yazılmış belgeleri tanıyan ve İngilizce Çince gibi metinlere çeviri desteği sağlayan otomatik bir sistem ele almıştır [12].

Teofilo de Campos ve arkadaşları İngilizce ve Kanada karakterleri içeren Hindistan Bangalore şehrinin sokak sahneleri görüntülerinden oluşan bir veri tabanı kullanarak OCR sistemi geliştirmiştir [13].

Çetiner ve arkadaşlarının yaptığı çalışma kamera ile alınan kimlik görüntülerinden, kimlik numaralarının hızlı şekilde tanımlanması ve gerçek zamanlı şekilde kişi bilgilerinin veritabanından getirilmesini içermektedir [14].

Rodriguez ve arkadaşları çalışma olarak yolcular için akıllı telefon ile çekilen işaret levhası görüntülerindeki İngilizce ifadeleri tanıyan ve bu ifadeleri Telegu diline çevirebilen, son olarak çevrilmiş metni ekranda gösteren bir web OCR sistemi geliştirilmiştir [15].

1. ÇALIŞMA

Projenin amacı ve kapsamı makine yazısı ile oluşturulmuş belgelerden metinsel verileri elde etmek için OCR görevlerinde yaygın olarak kullanılan 2 farklı yaklaşım olan Google Text Recognition ML Kit API ve derin öğrenme yöntemlerini kullanarak farklı ocr sistemleri geliştirilmesidir.

Çalışma yöntem farklılıkları açısından 2 alt başlık altında toplanacaktır. İlk başlıkta çalışmada kullanılan Google Text Recognition ML Kit API hakkında detaylı bilgiler sunulacaktır. İkinci başlıkta ise çalışmada kullanılan yapay sinir ağı mimarisi kapsamlı şekilde ele alınacaktır.

1.1 Google Text Recognition API ile Optik Karakter Tanıma

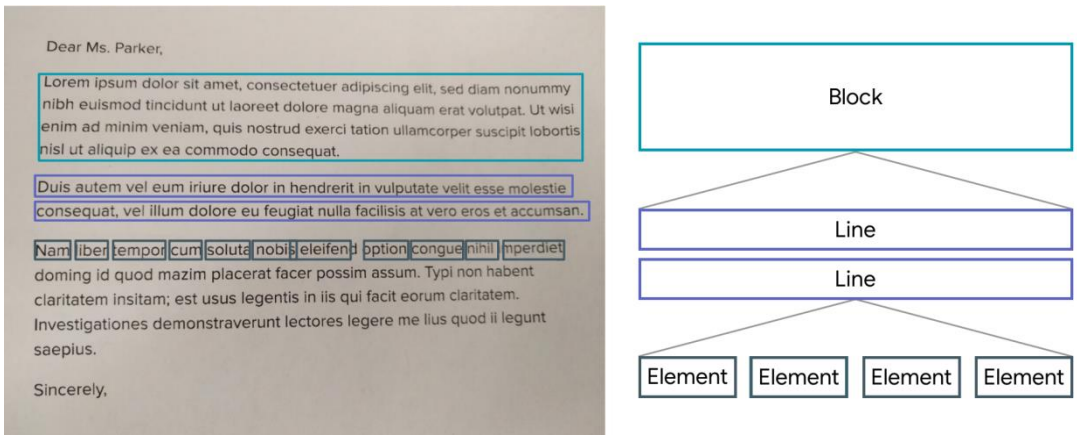
Bu bölümde, Google Text Recognition ML Kit API kullanarak bir OCR sistemi geliştirilmesi süreci detaylı olarak açıklanmaktadır. Google Text Recognition ML Kit API, Google tarafından sağlanan ve metin tanıma görevleri için optimize edilmiş bir yazılım kütüphanesidir. Bu API, görüntülerdeki metinleri tanıyarak dijital metin formatına dönüştürme görevini üstlenir. Çalışma görüntülerdeki Türkçe karakterlerin tanınması ve oluşturulan metnin ekranda gösterilmesi işlevlerini üstlenmektedir. Android platformunda geliştirilerek mobil cihazlarda etkinlik göstermesi hedeflenmiştir.

1.1.1 Google Text Recognition ML Kit API

Google Text Recognition ML Kit API Google tarafından video ve görüntülerden metin tanıma için sunulan arayüzdür. Bu arayüz Latin karakterleri, Çince, Korece, Devenegari, Japonca gibi bir çok dilde metin tanıma desteği sağlamaktadır. Metin tanıma sonrasında metnin ait olduğu dili de tanımlar. Banka kartı bilgileri, kartvizitlerin işlenmesi gibi veri girişi görevlerini doğru ve hızlı şekilde gerçekleştirmek için de kullanılabilmektedir.

Temel olarak metin tanıma için yüklenen görüntüyü paragraf, satır, kelime bazında analiz eder. Metni segmentasyon ile şekil1'de gösterildiği gibi bloklara ayrılır.

Bloklar bitişik metin satırı kümesidir. Bloklar parçalanarak Satırları oluşturur. Satır aynı eksen üzerinde yer alan bitişik kelime kümesidir. Satırlar da son aşamada parçalandığında öğeler elde edilmektedir. Öğe aynı eksende yer alan ardışık alfanümerik karakterlerdir. Bu bağlamda algılanan tüm öğeler API tarafından köşe noktaları , sınırlayıcı kutuları, döndürme bilgileri, güven puanı, ve tahmin edilen metni ile birlikte döndürülür [7].



Şekil 2.1. ML Kit ile Metin Tanıma Süreci

1.1.2 Android Platformu

Android Google tarafından geliştirilmiş ve mobil cihazlar için tasarlanmış işletim sistemidir. Temel özellikleri bağlamında Android işletim sistemi açık kaynaklıdır. Geliştiricilerin işletim sistemi yazılımını incelemeleri ve değiştirmelerini mümkün kılmaktadır. Kullanımını dünya çapına yaymak ve reklamlarının görünürlüğünü arttırmak amacıyla ücretsiz olarak kullanıma sunulmaktadır [16]. Android, linux çekirdeği üzerinde inşaa edilmiştir [16]. Mobil cihaz, akıllı saat, tablet, televizyon gibi çeşitli donanımlara uyum sağlamaktadır[16].

Bu çalışmada mobil cihazlar için metin tanıma uygulaması amacıyla Android platformu kullanılacaktır. Android uygulamasını geliştirmek için Android Studio IDE kullanılmıştır. Uygulama Java programlama dilinde yazılmıştır. Google Text Recognition ML Kit API 21'in altındaki API düzeyini desteklememektedir [7]. Bu nedenle çalışmada minSdkVersion değeri 31 tanımlanmıştır.

1.1.3 Çalışma Metodolojisi

Geliştirme süreci noktasında öncelikle proje kurulumu için Android Studio'da proje oluşturulur ve ML Kit Android kitaplıklarının bağımlılıkları eklenir. Daha sonra metin tanıma için giriş görüntüsü hazırlanır.Giriş görüntüsü kamera veya galeriden alınabilmektedir. Alınan görüntü ile InputImage nesnesi oluşturulur. Sonrasında metin tanıma işlemi gerçekleştirilir. InputImage nesnesi TextRecognizer sınıfının process yöntemine geçirilerek işlenen görüntü üzerinde metin tanıma algoritmaları çalıştırılır ve metin tanıma işlemi gerçekleştirilir. Son olarak metin tanıma başarılı bir şekilde tamamlanırsa resimde tanınan metin kullanıcıya döndürülür.

1.2 Yapay Sinir Ağları ile Optik Karakter Tanıma

Bu bölüm yapay sinir ağları kullanılarak geliştirilen OCR (Optik Karakter Tanıma) sistemi üzerinde yapılan çalışma hakkında bilgi vermektedir. Streamlit tabanlı uygulama, görsel işleme ve yapay sinir ağları kullanılarak metin bölgelerini tanımlar ve her bir kelime için optik karakter tanıma modelini kullanarak metinleri çıkarır. Elde edilen metinler, kullanıcıya görsel arayüz üzerinden sunulur. Bu çalışmada, geliştirilen OCR (Optik Karakter Tanıma) sistemi Streamlit frameworkü kullanılarak implemente edilmiştir. Uygulama, kullanıcıların bir görsel seçmelerine ve seçtikleri görsel üzerinde metin tanıma işlemleri yapmalarına imkan vermektedir. Kullanıcılar, uygulamaya bir görsel yüklemek için dosya yükleme aracını kullanır. Yüklenen görsel geçici olarak bir dosyaya kaydedilir. Yüklenen görsel, işlenebilir formata dönüştürülür. Önişleme adımları arasında görüntü büyütme, gri tonlama, ve ters-beyaz dönüşüm gibi işlemler bulunur. Ön işlenmiş görsel üzerinde yatay ve dikey histogram projeksiyonları kullanılarak kelime bölgeleri tespit edilir. Her kelime için bir bounding box oluşturulur ve bu kutular içindeki kelimeler ayrıştırılır. Her bir kelime bölgesi, eğitilmiş bir OCR modeli kullanılarak metin olarak tanımlanır. Bu adımda, kullanılan OCR modeli, görseldeki her bir kelimeyi tanımlamak için kullanılır. Her bir kelime için elde edilen metin tahminleri, kullanıcıya görsel arayüz üzerinde gösterilir.

2. YÖNTEM

Bu bölümün ilk çalışması Android Studio kullanılarak geliştirilmiş bir mobil uygulama üzerinde odaklanmaktadır. Uygulama, kullanıcıların galeriden veya doğrudan kamera kullanarak bir görüntü seçmelerine olanak tanır ve seçilen görüntü üzerinde metin algılama işlemleri gerçekleştirir. Metin algılama işlemi, Google ML Kit API kullanılarak yürütülmektedir, bu da uygulamanın hızlı ve etkili bir şekilde metin tanıma sağlamasını sağlar.

2.1 Google Text Recognition ML Kit ile Optik Karakter Tanıma

Şekil 3.1. Görüntüden Metne Dönüşüm İşlemi

2.1.1 Geliştirme Ortamı ve Araçları

Bu çalışma, Android Studio ortamında gerçekleştirilmiştir. Android Studio Giraffe | 2022.3.1 Patch 2 sürümü kullanılmıştır. Uygulama geliştirme sürecinde Java programlama dili tercih edilmiştir. Google ML Kit'in OCR özelliği, metin tanıma işlemlerini gerçekleştirmek için kullanılmıştır. Çalışma ortamı Windows 11 işletim sistemi üzerinde kurulmuştur ve geliştirme sırasında OpenJDK 64-Bit Server VM by JetBrains s.r.o. kullanılmıştır.

Projede açık kaynak (Open Source) kodlu bir yapım aracı (Build Tool) olarak Gradle kullanılmıştır [19]. Gradle yazılım geliştirme süreçlerinde kullanılan bir inşa (Build) aracıdır. Yeni bir proje oluşturulduğunda Gradle devreye girerek projeyi inşa eder[20]. Build işlemi projenin uygulama kaynaklarını ve kaynak kodunu bir araya

getirerek, test edilebilir, uygulanabilir, imzalanabilir ve nihayetinde yayınlanabilir bir formatta (APK veya AAB) oluşturmasıdır[20]. Proje Gradle altyapısıyla oluşturulduğu için 2 tür build.gradle dosyası yer almaktadır. Bunlar proje düzeyinde ve modül düzeyinde olarak ayrılmaktadır. Proje düzeyindeki "build.gradle" dosyası proje bazında yapılandırma ve bağımlılıkları içermektedir. Modül düzeyinde "build.gradle" dosyası uygulamanın SDK sürümleri, bağımlılıkları gibi uygulama düzeyinde yapılandırma ve bağımlılıklarını içermektedir.

2.1.2 Proje Yapılandırılması

Bu bölüm projenin aşamalarını ele almaktadır.

2.1.2.1 Yeni Proje Oluşturulması

Android Studio'da boş bir etkinlik (Empty Activity) şablonu kullanılarak yeni bir proje oluşturulmuştur. Proje oluşturulurken, minimum API seviyesi olarak API 31 seçilmiştir.

2.1.2.2 Açılış Ekranı (Splash Screen) Oluşturulması

Bu başlık, açılış ekranını oluşturan temel unsurları içerir. Açılış ekranı mobil uygulama yüklendiğinde görüntülenerek uygulamanın logosu, adı, veya açılış mesajı gibi kullanıcıya gösterilmek istenen bilgilerin yer aldığı tanıtım ekranıdır [22]. Kullanıcı uygulamayı açtığında geçici bir süre gösterilerek ana ekrana geçiş yapılır [21].

2.1.2.2.1 Tasarım

Açılış ekranı tasarımı için "introscreen.xml" adında bir XML dosyası oluşturulmuştur. Açılış ekranı 3 temel yapı içerir. Bunlar ConstraintLayout, ImageView ve TextView'dır.

ConstraintLayout ile öğeler arasındaki bağlantı ve hizalamalar yönetilmiştir. Bu sayede yerleştirilen öğelerin hangi öğelerle birlikte hareket edebileceği ve üstten, alttan, soldan ve sağdan nasıl hizalanacağı tasarlanır.

ImageView açılış ekranında gösterilecek uygulama logosunu temsil etmektedir. Burada kullanılan logo introLogoImage olarak adlandırılmış ve bir drawable kaynağı (`@drawable/im0077782`) ile ilişkilendirilmiştir. Boyutları belirlenmiş ve üstten ve soldan 200dp'ye kadar hizalanmıştır.

Textview ise açılış ekranına yerleştirilen metin öğesini temsil etmektedir. Burada kastedilen metin öğesi uygulamanın ismidir. `introText` olarak adlandırılmış ve `@string/easy_ocr` dize kaynağı ile ilişkilendirilmiştir. Metin rengi mavi (`@color/BLUE`) olarak belirlenmiş, metin boyutu 34sp ve kalın (`bold`) olarak stilize edilmiştir. `introLogoImage` öğesinin altında ve soldan 32dp uzaklıkta yer alır.

Şekil 3.2. Açılış Ekranı Tasarım Sayfası

2.1.2.2.2 İşlevsellik

Bu ekran uygulamanın başlangıcında kullanıcının bir süre splash screen ekranını görmesini sağlar ve ardından ana sayfaya geçişi gerçekleştirir. Geçişi sağlamak için Handler sınıfı kullanılır. Bu sınıf, bir fonksiyonun ne kadar süre ile

gerçekleştirileceğini tanımlar [17]. Uygulamada gecikme süresi değeri 2 saniye olarak belirlenmiştir. Aktivite başlatıldığında tanıtım ekranı tasarımını içeren layout ekrana getirilir. Bu süre içinde uygulamanın ismini içeren ActionBar gizlenir. Uygulama 2 saniye ekranda kalır ve sonrasında Ana sayfaya geçiş gerçekleştirilir.

2.1.2.3 Ana Sayfa (Main Page) Oluşturulması

Bu bölümde, kullanıcıların bir görüntüden metin algılamalarını ve bu metni kopyalamalarını veya silmelerini sağlayan ana sayfa ekranı açıklanmaktadır. Görüntüden metin algılama özelliği, kullanıcıların kolay arayüz üzerinden kolayca metin içeriği çıkarmalarına olanak tanır.

2.1.2.3.1 Tasarım

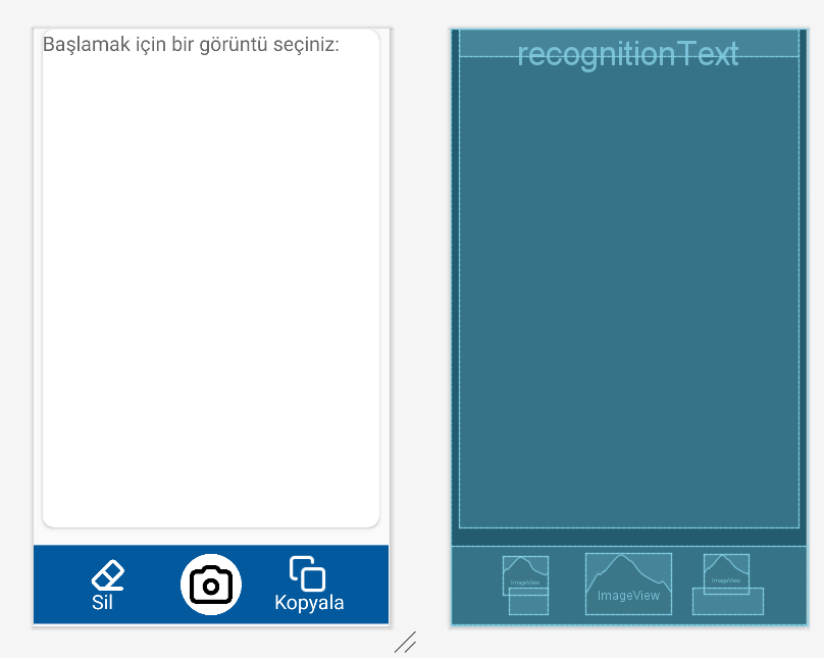
Bu bölümde Android uygulamasının ana sayfasını oluşturan UI bileşenlerinin düzenlenmesi ele alınmaktadır. Layout, başlık, görüntü gibi UI bileşenlerinin düzenlenmesidir [23]. Ana sayfa tasarımında 2 ana layout kullanılmaktadır. Biri tüm sayfa için, diğeri ise alt menü için kullanılmıştır.

Bu tasarımda layout olarak bileşenlerin birbirine ve ekrana göre nasıl konumlandırılacağını belirlemek için kısıtlamalar kullanarak esnek ve karmaşık düzenler oluşturmayı sağlayan ConstraintLayout kullanılmaktadır. Constraints, UI bileşenlerinin düzenlenmesini içeren kurallara sahiptir [24].

Üst kısımda, kullanıcıdan metin girişi alınmaktadır. Bu bölümde, `CardView` ve `EditText` bileşenleri kullanılmıştır. `CardView`, oluşturulan alanın köşelerinin yuvarlatılarak kart görünümü almasını sağlar. `CardView` içinde yer alan `EditText` bileşeni ise metin girişi almaktadır ve `hint` özelliği ile kullanıcıya bir görüntü seçmesi gerektiğini bildiren bir geri bildirim sunar.

Alt menüde, metinle ilgili işlemleri gerçekleştiren düğmeler bulunur. Kullanıcı deneyimini artırmak amacıyla, düğmelerin altında açıklayıcı metinler yer almaktadır. Alt menü için ayrı bir `ConstraintLayout` oluşturulmuştur. Bu düzenleme, alt menünün arka plan rengi, boyutu ve hizalaması gibi kısıtlamaları belirlemektedir. Alt menü içerisinde, silme, kopyalama ve görüntü seçme işlevleri için düğmeler ve

açıklayıcı metinler bulunmaktadır. Bu layout içinde yer alan `ImageView` bileşenleri, silme, kopyalama ve görüntü seçme düğmelerinin ikonlarını göstermekte, `TextView` bileşenleri ise düğmelerin işlevlerini açıklamak için kullanılmaktadır.



Şekil 3.3. Ana Ekran Tasarım Sayfası

2.1.2.3.2 İşlevsellik

Kullanıcılar cihazlarının galerisinden veya doğrudan kamera ile bir görüntü seçerek işleme başlayabilirler. Bu, kullanıcıların uygulamayı çeşitli görüntü kaynaklarına entegre etmelerini sağlar. Seçilen görüntü üzerindeki metin algılanır ve kullanıcıya sunulur. Bu Google ML Kit API kullanımı ile gerçekleştirilir. Algılanan metin kullanıcının panosuna kopyalanabilir. Bu, kullanıcıların metni başka uygulamalarda kullanmak veya saklamak için kolayca kopyalamasına olanak tanır. Algılanan metin, kullanıcı tarafından silinebilir. Bu, kullanıcıların temiz bir işlem alanı oluşturmasını sağlar ve yanlışlıkla algılanmış veya gereksiz metinleri kaldırmasına olanak tanır. Uygulama metin algılama işlemi sırasında oluşabilecek hataları ele alır ve

kullanıcıya bildirir. Bu, kullanıcıların işlem sırasında bilgilendirilmelerini ve hata durumlarında uygun şekilde yönlendirilmelerini sağlar.

2.2 Yapay Sinir Ağları ile Optik Karakter Tanıma

Yapay Sinir Ağları ile geliştirilen OCR sistemi, görüntü işleme ve metin tanıma süreçlerini adım adım içermektedir. İlk olarak, kullanıcıların JPEG veya PNG formatındaki görüntü dosyalarını yükleyebilmesi için Streamlit kullanılmıştır. Ardından, görüntüye keskinleştirme filtresi uygulanarak metin bölgeleri daha belirgin hale getirilmiştir. İkili formata dönüştürme adımında, gri tonlamalı görüntü üzerine Otsu'nun ikili eşikleme yöntemi uygulanarak metin bölgeleri net bir şekilde tespit edilmiştir. Morfolojik işlemler, görüntüdeki metin bölgelerini daha da belirginleştirmiş ve arka plandaki gürültüyü azaltmıştır.

Yatay ve dikey histogram projeksiyonları, sırasıyla satırların ve kelimelerin tespit edilmesinde kullanılmıştır. Yatay projeksiyon, metin satırlarının başlangıç ve bitiş noktalarını belirlemiş ve satır segmentasyonunu gerçekleştirmiştir. Dikey projeksiyon ise her bir satır içindeki kelime bölgelerini tespit ederek kelime segmentasyonunu sağlamıştır. Son olarak, her bir kelime bölgesi, özel olarak eğitilmiş OCR modeline verilerek metin tanıma işlemi yapılmış ve tahmin sonuçları kullanıcıya sunulmuştur. Bu adımlar, geliştirilen OCR sisteminin görüntüler üzerinde başarılı bir şekilde metin tanıma işlemi gerçekleştirdiğini göstermektedir.

2.2.1 Geliştirme Ortamı ve Araçları

Projede bilgisayarlı görü görevlerinde sıkça kullanılan OpenCv , hesaplama ve matris işlemleri için Numpy, uygulama oluşturulması için Streamlit, sonuçların görselleştirilmesi için Matplotlib, model eğitimi için Tensorflow kütüphanelerinden yararlanılmıştır.

2.2.2 Proje Yapılandırılması

Proje Visual Studio Code ortamında yazılarak test edilmiştir. Python programlama dili kullanılarak gerçekleştirilmiştir. Streamlit platformunda uygulama haline getirilmiştir.

3.2.2.1 Görüntü Önişleme

Önişleme, görüntünün kullanım amacına yönelik performansını arttırmak amacıyla iyileştirmelere tabi tutulmasıdır [25]. Bu çalışmada yapılan önişleme adımları metin segmentasyonu ve karakter tanıma sürecinin daha doğru ve verimli olmasını sağlar. Bu bölümde önişleme aşamaları ele alınacaktır.

2.2.2.1.1 Görüntünün Yüklenmesi

İlk olarak, işlem yapılacak görüntü yüklenir. Bu adım, görüntünün boyutlarını değiştirmeden, üzerinde yapılacak işlemlere hazırlık aşamasıdır.

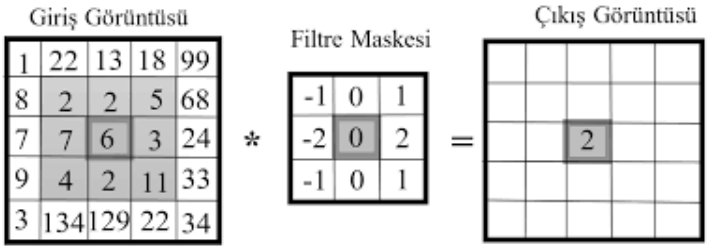
2.2.2.1.2 Görüntünün Keskinleştirilmesi

Keskinleştirme, görüntüdeki ince detayların vurgulanması amacıyla gerçekleştirilen işlemlerdir[27]. Çekilen görüntülerin farklı ışık veya açılarda çekilmesi gibi karakterlerin bulanık gözükmesine neden olmaktadır. Metin tanıma için karakterlerin netliği önem taşır. Keskinleştirme, detayları daha belirginleştirdiği için harflerin daha net ve belirgin olmasını sağlayarak karakterlerin doğru tanınmasını sağlar.

Keskinleştirme işlemi filtreleme ile yapılabilmektedir. Filtreleme yönteminde filtre tanımlanmalıdır. Filtre aslında çekirdek(kernel) matristir [26]. Genellikle 3x3 boyutunda tanımlanır. Bu çalışmada da 3x3 boyutunda bir çekirdek matris tanımlanmıştır. Çekirdek matrisin elemanlarının değeri keskinleştirmenin şiddetini belirlemektedir. Bu çalışmada merkezi değeri +10 çevresel değerleri -1 olmak üzere çekirdek matris tanımlanmıştır.

Filtreleme işlemi bir konvolüsyon işlemidir. Çekirdek matris görüntü üzerinde gezdirilerek her seferinde matris ile eşleşen piksellerin değerleri çarpılır, çarpım sonuçları toplanır, bu toplam matrisin elemanlarının toplamına bölünerek merkezi

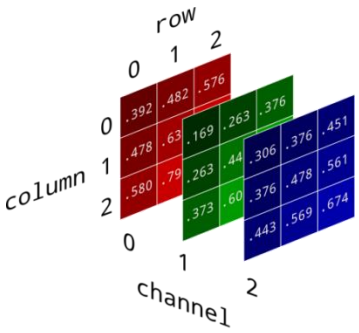
pikselin yeni değeri elde edilir[26]. Bu işlem her seferinde görüntünün bir pikselini etkileyerek sırasıyla tüm pikseller için uygulanır[26].



Şekil 3.4. Filtreleme İşlemi

2.2.2.1.3 Görüntünün Gri Tonlamalı Formata Dönüşümü

Renkli görüntü kırmızı, yeşil, mavi olmak üzere 3 seviye gri tonlamalı kanal içeren görüntülere denir [31]. Her kanal genellikle 8 bit ile gösterilir ve 0 ile 255 arasında değerler içerir. Bu nedenle RGB görüntüler 24 bit ile gösterildiğinden (0,0,0) ile (255,255,255) arası değerler almaktadır.



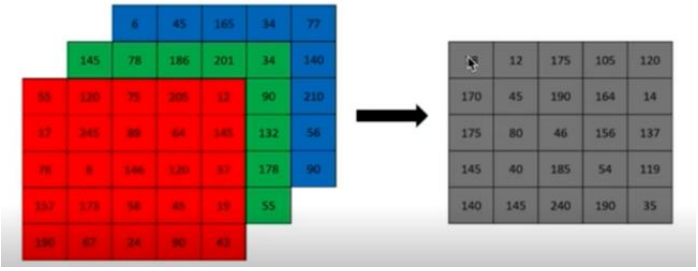
Şekil 3.5. RGB Resim Kanalları Gösterimi [29]

Gri tonlamalı görüntü beyaz, siyah ve bu 2 rengin arasındaki gri tonlarını içeren tek kanallı görüntüye denir [31]. Genellikle 8 bit ile kodlanırlar ve 0 ile 255 arasında değerlere sahip olurlar.

Renkli görüntüden gri tonlamalı görüntüye dönüşüm birçok açıdan önişlemede tercih edilmiştir. RGB görüntü 3 kanal içerirken gri tonlamalı görüntü tek kanal içermektedir. RGB'de görüntü renk bilgisine sahipken gri tonlama sadece piksel

yoğunluğu bilgisini tutar. Bu nedenle daha basit bir yapıdadır. Basit olması hesaplamaların daha hızlı yapılmasını ve bellekte kaplanan alanın daha az olmasını sağlar.

Bu aşamada RGB görüntünün gri tonlamaya dönüştürülülmesinde izlenen yöntem. her pikselin kırmızı, yeşil, mavi renk değerinin renk ağırlığı ile çarpılıp sonuçların toplanmasıdır. Elde edilen değer pikselin gri tonlamalı değeridir. Burada renk ağırlığı, insan gözünün renklere olan hassasiyetine dayanmaktadır.

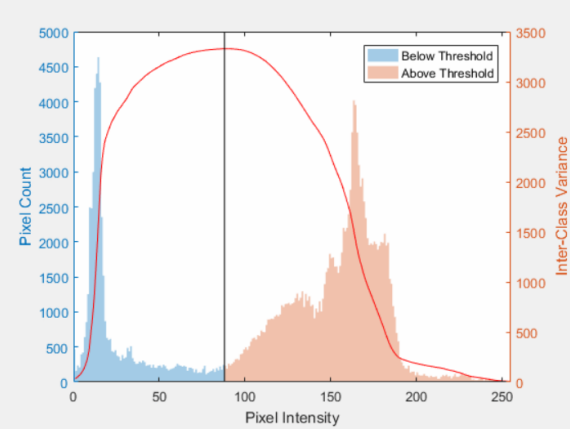


Şekil 3.6. Resmin RGB'den Gri Tonlamaya Dönüştürülmesi [30]

2.2.2.1.4 Görüntünün İkili Formata Dönüştürülmesi

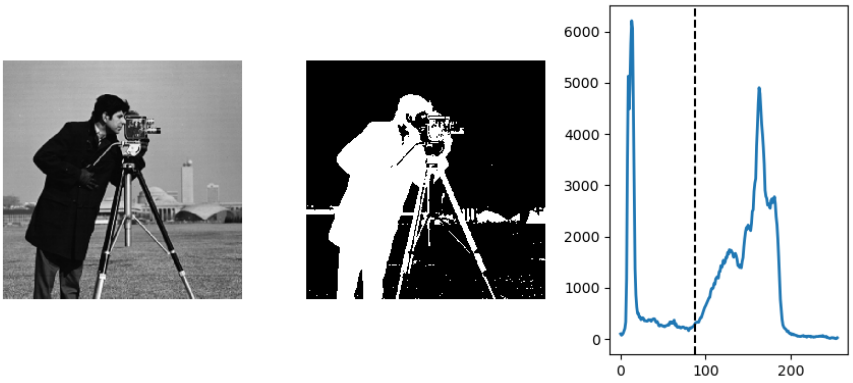
Bu aşamada görüntünün siyah ve beyaz şeklinde iki seviyeye dönüştürülmesi amaçlanmaktadır. İkili görüntüler beyaz ve siyah olmak üzere 2 çeşit renk değerine sahip görüntülerdir [32]. Bu bölümde görüntünün ikili formata dönüştürülmesi için ters ikili eşikleme yöntemi kullanılmıştır. Ters ikili eşikleme yönteminde eşik değerinin altında kalan pikseller beyaz ve üstünde kalan pikseller siyah renk olarak tanımlanır. Ters ikili eşikleme yönteminde sabit bir eşik değerini kullanılması her görüntüde doğru sonuç vermemektedir. Bu nedenle her görüntü için kendi piksel değerlerine en uygun olan optimal bir eşik değeri seçimi gerçekleştirilmelidir. Bu gibi sebeplerden dolayı çalışma bağlamında eşik değerinin seçilmesi için Otsu eşikleme yöntemi seçilmiştir. Bu yöntem gri seviyeli görüntüler üzerinde varyans kullanılarak adaptif bir eşik değeri seçilmesine dayalı olan eşik belirleme yöntemidir [33]. Bu yöntemde görüntüyü eşik değerinin üstünde kalan pikseller ön plan ve eşik değerinin altında kalan pikseller arka plan olacak şekilde 2 sınıfa ayırmak amacıyla eşik değeri belirlenir [34]. Bu eşik değeri sınıf içi yoğunluk varyansı minimize edilerek veya sınıflar arası yoğunluk varyansı maksimize edilerek bulunur [35].

Otsu yöntemi birtakım sınırlamalara sahiptir. Görüntünün gürültü seviyesi bunlardan biridir. Bu noktada gürültünün çok olması çizilen histogramın şeklini etkilemektedir. Histogram üzerinde sınıflar arası varyansı maksimize eden eşik değeri seçilmesi noktasında hatalı eşik değeri belirlenmesine neden olmaktadır. Bir diğer etken ışık değerinin dengesidir. Görüntüde ışığın dengesiz dağılımı, daha parlak ve daha karanlık bölgeler yaratacaktır. Bu nedenle histogramda da geniş ve dağınık uç noktaları yer alacağı için optimal eşik değeri belirlemek zor olabilmektedir [35].



Şekil 3.7. Otsu Eşikleme Yöntemi

Amaç bağlamında bu aşama histogram analizi gerçekleştirerek metin karakterleri ve arka plan olmak üzere görüntüyü 2 sınıfa ayırır. Bu aşama segmentasyon için temel oluşturmaktadır. Çünkü segmentasyonda görüntünün kelimelere bölünmesi için karakter sınırlarının net olması önem taşır. Bu aşamada arka plan ve karakterler arasındaki yüksek kontrast karakterlerin sınırlarının net olmasını sağlamaktadır. Böylece kelimeler daha iyi bölümlenmektedir.

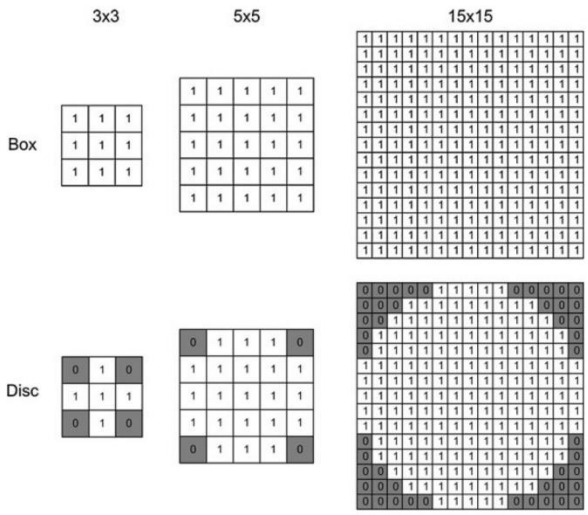


Şekil 3.8. Otsu Eşikleme Yöntemi İle İkili Resme Dönüşüm

2.2.2.1.5 Görüntüye Morfolojik İşlemler Uygulanması

Bu aşamada ikili görüntü üzerinde morfolojik işlemler gerçekleştirilir. Morfolojik işlem temel olarak imgenin yapısı ile ilgili işlemdir. Morfolojik işlemler yapısal eleman (Structuring Element) kullanılarak gerçekleştirilir. Gri tonlu veya ikili görüntülere uygulanabilmektedir. İmgenin yapısının analizi, sınırlarının çıkartılması, segmentasyon, gürültünün azaltılması gibi amaçlarla gerçekleştirilmektedir [36]. Bu çalışmada kelimelerin sınırlarının daha iyi belirlenmesi ve segmentasyon amacıyla önce genişletme, ardından erozyon işlemi uygulanmaktadır.

Morfolojik işlemler için yapı elemanı bir diğer adıyla çekirdek (kernel) kullanılmaktadır[37]. Yapı elemanı, görüntü üzerinde gezerek görüntüyü analiz etmeye yarayan matristir. Yapı elemanı, bir görüntü üzerinde hareket ederek imgelerin morfolojik özelliklerini değiştirmek amacıyla kullanılır. Yapı elemanının şekli imgelere uygulanacak morfolojik değişikliklerin nasıl yapılacağını belirler. Bu aşamada yapı elemanı şekli dikdörtgen seçilerek yapılan işlemin belirli yönlerden ziyade tüm yönlerde eşit şekilde gerçekleştirilmesi sağlanmıştır. Yapı elemanının büyüklüğü işlemin etkisini belirlemektedir. Bu çalışmada 3x3 boyutunda yapı elemanı kullanılmaktadır. Yapı elemanları ayrıca farklı komşuluk boyutlarında tanımlanmaktadırlar [38].



Şekil 3.9. Morfolojik Filtreler

2.2.2.1.5.1 Genişletme (Dilation)

İlk olarak genişletme (dilation) aşaması uygulanmaktadır. Genişletme mantıksal olarak ele alınan nesnenin sınırlarının genişletilmesine dayanmaktadır. Ön plan pikselleri büyüyeceği için genişlemekte ve boyutu artmaktadır [38].

Çalışmada kullanılan ikili görüntüde metin karakterleri beyaz ve arka plan siyah piksellerden oluşmaktadır. Metin karakterleri 2 tür boşluk içermektedir. İlk tür boşluk kelimeler arasında ve kelime sınırlarını belirten boşluktur. İkinci tür ise kelimelerin içinde harfler arasında yer alan boşluktur. Görüntüde kelimelerin bölütlenmesi için kelimeleri oluşturan harflerin pikselleri genişleyerek harfler arasındaki boşluklar azaltılmalıdır. Bu şekilde kelimeler bir bütün halinde daha kolay bölütlenmektedir. Bu nedenle genişletme sayesinde kelimeleri oluşturan karakterler yani beyaz pikseller büyütülür. Karakterler birbirine yaklaşarak aralarındaki boşluklar yani siyah pikseller azalmaktadır. Kelimelerin sınırları daha net belirlenmektedir. Bu işlem, nesneler arasındaki boşlukları doldurarak ve nesnelerin

kenarlarını birleştirerek metin bloklarını daha birleşik hale getirir. Bu durumda segmentasyon daha verimli gerçekleştirilmektedir.

Şekil 3.10. Genişletme İşlemi

Genişletme işleminde yapı elemanı kullanılmaktadır. Bu aşamada yapı elemanı 3x3 boyutunda ve dikdörtgen şeklinde tanımlanmıştır. Dikdörtgen tanımlanması genişletmenin her yöne eşit şekilde yapılacağını belirlemektedir. Genişletmenin iterasyon sayısı 4 olarak belirlenmiştir. Bu nedenle genişletme 4 kez tekrarlanarak etkisi arttırılmıştır.

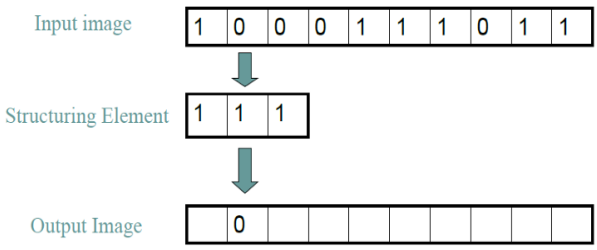
2.2.2.1.5.2 Aşındırma (Erosion)

İkinci olarak aşındırma uygulanmaktadır. Aşındırma mantıksal olarak ele alınan imgenin sınırlarının aşındırılmasına bir diğer deyişle küçültülmesine dayanmaktadır. Ön plan piksellerini küçülterek nesneyi küçültmektedir. Küçültmek için beyaz pikseller çevresindeki siyah piksellerle yer değiştirmektedir.

$$A \ominus B = \left\{ z \middle| (B)_z \subseteq A \right\}$$

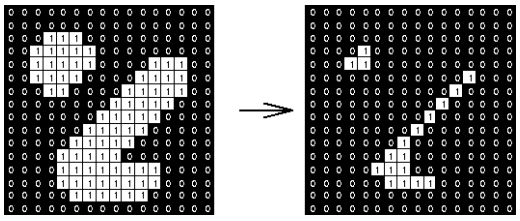
Aşındırma operatörü

Şekil 3.11. Aşındırma İşlemi Matematiksel Formülü



Şekil 3.12. Matrise Aşındırma İşlemi Uygulanması

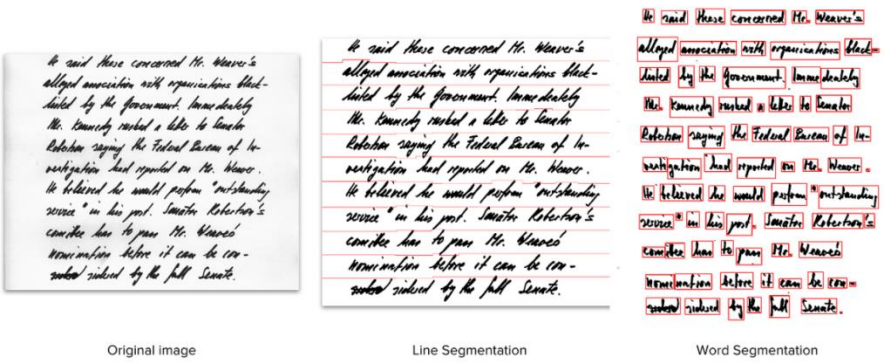
Bu işlem, metin bölgelerini küçültür ve ince detayları ortadan kaldırır. Genişletme işleminin ardından uygulanan aşındırma istenmeyen gürültüyü azaltmış olur. Bu işlem, nesnelerin kenarlarını düzeltir ve gereksiz detayları temizleyerek metin bloklarını daha keskin hale getirir. Çalışmada aşındırma işlemi için dikdörtgen ve 3x3 boyutunda yapı elemanı kullanılmıştır. Aşındırma işlemi 3 kez uygulanarak etkisi arttırılmıştır.



Şekil 3.13. Görüntüye Aşındırma Uygulanması

2.2.2.2 Segmentasyon

Segmentasyon ele alınan görüntünün daha iyi analiz edilebilmesi için anlamlı birimlere bölünmesidir. Segmentasyon optik karakter tanıma görevlerinde satır, kelime ve karakter gibi 3 düzeyde metin parçalarını tespit etmek için kullanılmaktadır. Segmentasyon için eşikleme, kenar tespiti, bölge tabanlı yöntemler gibi yaygın kullanılan bölütleme teknikleri vardır. Bu çalışmada bölge tabanlı yöntemlerin bir alt dalı olan Histogram Projeksiyonu tekniği ile segmentasyon uygulanmaktadır.



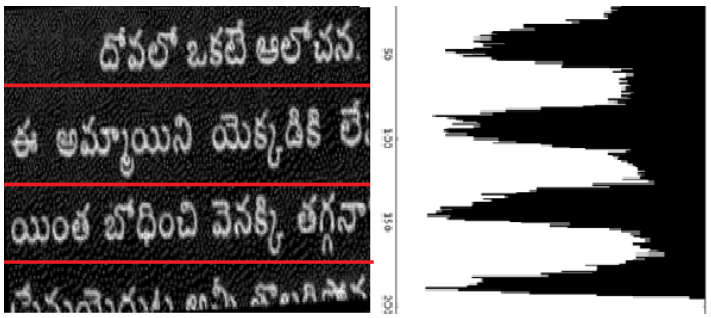
Şekil 3.14. Satır ve Kelime Düzeyinde Segmentasyon

2.2.2.2.1 Histogram Projeksiyonu

Histogram projeksiyonu görüntü işleme alanında kullanılan bir tekniktir. Bu yöntem görüntü üzerinde belirli bir yön seçerek (dikey veya yatay) bu yönde piksel yoğunluğunu analiz etmek için kullanılır. İkili görüntü siyah ve beyaz olmak üzere 2 tür piksel içermektedir. Bu pikseller görüntüyü ön plan ve arka plan gibi 2 bölgeye bölmek için analiz edilir. Görüntüde ilgi duyulan bölgeyi temsil eden piksellere ön plan pikselleri denir [40]. İlgi duyulan bölge dışında kalan bölgelere arka plan pikselleri denir [40]. Görüntü ikili görüntü formatına dönüştürülürken belirli bir eşik değeri kullanılarak görüntü pikselleri ön plan ve arka plan piksellerine ayrılır. Bu çalışmada ikili görüntüdeki metin bölgeleri beyaz renk ve arka plan siyah renk olduğundan beyaz pikseller ön planı ve siyah pikseller arka planı temsil etmektedir. Histogram Projeksiyonu Yatay Histogram Projeksiyonu ve Dikey Histogram Projeksiyonu olmak üzere 2 çeşittir [41]. Bu çalışmada satır tespiti için Yatay Histogram Projeksiyonu ve kelime tespiti için Dikey Histogram Projeksiyonu kullanılmaktadır.

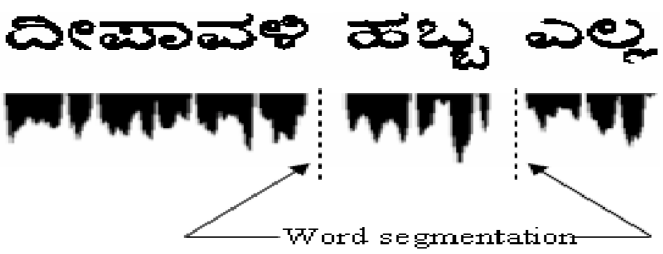
Yatay Histogram Projeksiyonu bir ikili görüntüdeki yatay eksen boyunca piksel yoğunluklarının hesaplanmasını sağlamaktadır. Bu çalışmada görüntü ikili görüntüye dönüştürüldüğünde metin karakterleri (satırlar) beyaz, arka plan ise siyah piksellere dönüştürülür. Görüntü üzerinde tespit edilmesi amaçlanan bölgeler satırların bulunduğu bölgeler olduğu için beyaz pikseller ön plan pikselleri ve siyah pikseller

arka plan pikselleri olarak belirlenir. Yatay eksende ön plan piksellerin yoğun olduğu bölgeler satırları içermektedir. Bu yoğun bölgeler tespit edilerek satırlar belirlenir. Yoğunluğun az olduğu bölgeler de satır aralarını temsil eder.



Şekil 3.15. Görüntüye Yatay Histogram Projeksiyonu Uygulanması

Dikey Histogram Projeksiyonu görüntüdeki dikey eksen boyunca ön plan piksellerinin sayısının hesaplanması işlemidir. Sonuç dizisi görüntünün sütun sayısı ile aynı boyutta olmaktadır. Yatay Histogram Projeksiyonu uygulanan görüntüdeki her bir satır ayrı bir görüntü olarak ele alınır. Sonrasında bu satırların dikey eksende ön plan pikselleri sayılır. Kelimeler ön plan piksellerinin yoğun olduğu ve kelime sınırları ön plan piksellerinin az olduğu alanlardır. Bu sayede satırlar bölümleme çizgileri ile kelimelere bölütlenmektedir.



Şekil 3.16. Görüntüye Dikey Histogram Projeksiyonu Uygulanması

2.2.2.2.2 Segmentasyon Seviyeleri

Segmentasyon 3 farklı seviyede gerçekleştirilebilmektedir. 1.Yöntem satır düzeyinde segmentasyondur. Bu yöntemde görüntü üzerindeki metin satır parçalarına ayrılır. 2. yöntem kelime düzeyinde segmentasyondur. Bu yöntemde metin görüntüsü kelime parçalarına ayrılır. 3. yöntem karakter düzeyinde segmentasyondur. Bu yöntemde metin görüntüsü karakter parçalarına ayrılır. Bu yöntemlerden hangisinin veya hangilerinin birlikte kullanılacağı gerçekleştirilecek göreve bağlıdır. Bu çalışmada satır ve kelime düzeyinde segmentasyon gerçekleştirilmiştir. Segmentasyon gerçekleştirmek için Histogram Projeksiyonu tekniği uygulanmıştır.

Satır Düzeyinde Segmentasyon görüntü üzerindeki metnin satırlarının tespit edilmesidir. Bu çalışmada satır düzeyinde segmentasyon uygulanarak metin görüntüsündeki satırlar tespit edilmektedir. Satır düzeyinde segmentasyon için Yatay Histogram Projeksiyonu tekniği uygulanmıştır. Belgede yer alan metin satırları piksel yoğunluğu yüksek alanlardır. Beyaz piksel sayısının satır bazında toplamı alınarak satırların başlangıç ve bitiş noktaları tespit edilir ve bir listede saklanır.

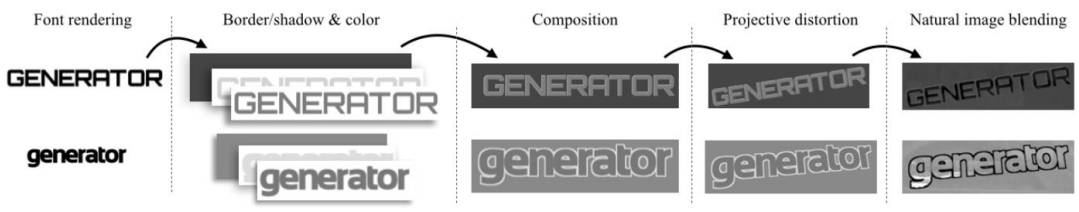
Kelime Düzeyinde Segmentasyon görüntü üzerindeki kelimelerin tespit edilmesidir. Bu aşamada kelime düzeyinde segmentasyon uygulanarak metin görüntüsündeki kelimeler tespit edilmektedir. Kelime düzeyinde segmentasyon için Dikey Histogram Projeksiyonu tekniği uygulanmıştır. Satır düzeyinde segmentasyonda elde edilen her satır kelime düzeyinde segmentasyona tabi tutulur. Belgede yer alan her satır için kelimeler piksel yoğunluğu yüksek ve kelimeler arasındaki boşluklar piksel yoğunluğu düşük alanlardır. Dikey Histogram Projeksiyonu ile beyaz piksel sayısının sütun bazında toplamı alınarak kelimelerin başlangıç ve bitiş noktaları tespit edilir. Son olarak, tespit edilen kelimelerin sınırlayıcı kutuları çizilir ve numaralandırılır. Sınırlayıcı kutular yukarıdan aşağıya ve soldan sağa sıralanarak çizilir.

2.2.2.2.3 Segmentasyon Sonuçlarının Kaydedilmesi

Bu çalışmada birden fazla satırdan oluşan uzun metinlerin tüm kelimelerinin doğru sıralama ile ve satır sütun düzenine uygun şekilde çıktısını elde etmek amacıyla önce satır düzeyinde daha sonra kelime düzeyinde segmentasyon gerçekleştirilmiştir. Dikey Histogram projeksiyonu ile satırlar bölütlendikten sonra satır koordinatları diziye eklenerek iterasyon ile her satıra Dikey Histogram Projeksiyonu uygulanmıştır. Bu yöntemle bölütlenen kelimeler kendi satırında 1'den başlayarak soldan sağa indekslenmiştir. Çıktı olarak elde edilen kelime görüntüleri ilgili klasöre gri tonlamalı formatta ve ilk sayı satır indeksini, ikinci sayı ise o satırdaki sütun indeksini gösterecek şekilde isimlendirilerek kaydedilmiştir.

2.2.2.3 Veri Seti

MJSynth Dataset, doğal sahne metin tanıma için sentetik olarak oluşturulmuş bir veri setidir. Bu veri seti, yaklaşık 10 GiB boyutundadır. Synthetic Word Dataset, gerçek dünya görüntülerinde metin tanıma için yeterli olan sentetik olarak üretilmiş bir veri setidir. Bu veri seti, farklı yazı tipleri, sınır ve gölge efektleri, renk değişiklikleri, kompozisyon varyasyonları, perspektif bozulmaları ve doğal görüntülerle harmanlama gibi çeşitli teknikler kullanılarak oluşturulmuştur. 9 milyon görüntü ve 90.000 İngilizce kelimeyi kapsayan bu veri seti, eğitim, doğrulama ve test bölümleriyle birlikte sunulmaktadır. Bu veri seti, derin öğrenme modellerini eğitmek ve test etmek için kullanılır ve metin tanıma araştırmalarında önemli bir kaynak olarak hizmet eder.



Şekil 3.17. SynthText Veri Seti

2.2.2.4 Model

2.2.2.4.1 Yapay Zeka

Yapay zeka literatürde “Artifical Intelligence” terimine karşılık gelmektedir. Yapay zekanın tanımı teknoloji ilerledikçe değişmektedir. Genel olarak yapay zeka bir bilgisayar sisteminin akıllı varlıklara özgü görevleri yerine getirebilme yeteneğidir. Akıllı varlıklar öğrenme, olaylar arasında ilişki kurma ve anlam çıkarma, deneyim kazanma, genelleme yapma gibi entelektüel niteliklere sahiptir. Yapay zeka insana özgü görevleri gerçekleştirerek süreçleri kolaylaştırmaktadır. Yapay zeka problemlerin çözümü, oyunların modellenmesi,doğal dil işleme, örüntü tanıma gibi farklı alanlarda uygulamalara sahiptir.

2.2.2.4.2 Makine Öğrenimi

Makine öğrenimi yapay zekanın alt dalıdır. Genel olarak makine öğrenimi bir bilgisayar sisteminin açıkça programlanmadan geçmiş deneyimlerinden öğrenerek genelleme yapabilmesi ve karar verebilmesidir. Bunun için birçok istatistiksel modeller ve algoritmalar geliştirilmiştir. Bu sayede algoritma verilerle beslenerek eğitilir ve öğrenme sürecine girer. Eğitildikten sonra genelleme yeteneği elde ederek daha önce görmediği yeni bir durumla karşılaştığında karar vermektedir. Bu sayede açık talimatlar verilmeden görevler gerçekleştirmektedir.

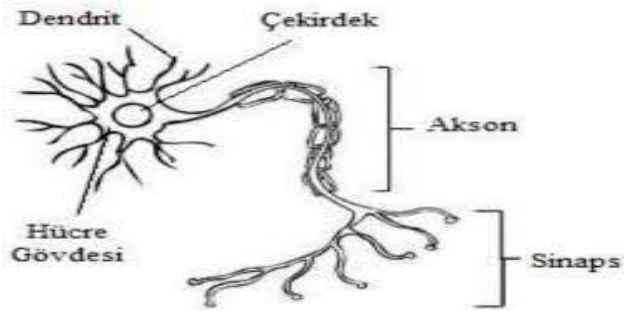
2.2.2.4.3 Yapay Sinir Ağı

Yapay sinir ağı (YSA) insan beyninin biyolojik sisteminin taklit edilmesine dayalı derin öğrenme tekniğidir. İnsan beyninde nöronlar ve bu nöronlar arasında bağlantıyı sağlayan sinaptik bağlar bulunmaktadır. Nöron ve sinaptik bağlarla oluşturulan ağda öğrenme, yeni bilgiler üretme, bu bilgileri hafızada saklama, veriler arasında ilişkiler kurma gibi işlevler gerçekleştirilir. YSA biyolojik sinir ağlarını taklit eden ve bir dizi katman aracılığıyla veri işleyen algoritmalardır.

2.2.2.4.3.1 Biyolojik Sinir Hücresi

Bir biyolojik sinir sistemi biyolojik sinir ağı bir diğer deyişle nöronlardan meydana gelmektedir. Nöronlar akson, dendrit, çekirdek, sinaps olmak üzere 4 bölümden

meydana gelmektedir. Dendritlerin temel işlevi bağlantı kurduğu diğer nöronlardan gelen sinyalleri çekirdeğe iletmektir. Çekirdek ise dendritten gelen sinyalleri toplayarak aksona gönderir. Akson çekirdekten aldığı sinyalleri işler ve sinapslara gönderir. Sinapslar ise sinyalleri diğer nöronlara iletir. Özetle duyu organı vb. aracılığıyla alınan sinyaller taşıyıcı nöronlar vasıtasıyla bir nörondan diğerine iletilerek merkezi sinir sistemine aktarılır. Burada sinyal işlenerek uygun tepki sinyali üretilir ve bu sinyal tepkinin oluşacağı duyu organlarına aktarılarak süreç sonlanır.

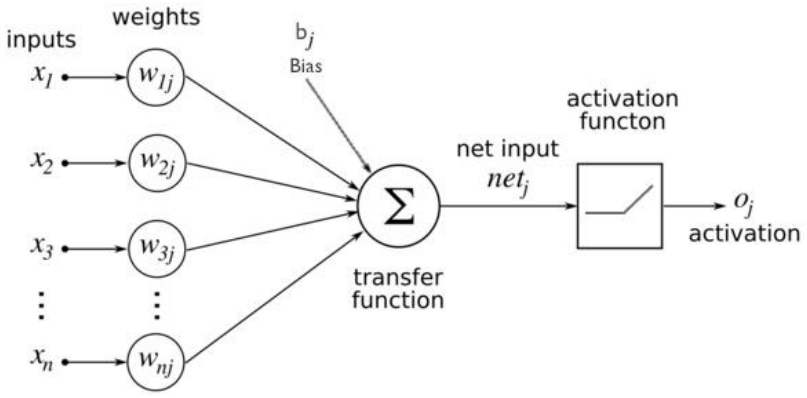


Şekil 3.18. Biyolojik Sinir Hücresinin Yapısı

2.2.2.4.3.2 Yapay Sinir Hücresi

Yapay sinir hücresi, aynı zamanda perceptron olarak da bilinir, yapay sinir ağlarının temel yapı taşıdır. Yapay sinir hücreleri biyolojik sinir hücrelerinden esinlenerek oluşturulduğu için benzer yapıdadırlar. Bu bağlamda biyolojik sistemdeki gibi nöronlar aralarında bağ kurarak ağlar oluştururlar. Biyolojik sistemlere benzer şekilde sinyaller nöron tarafından alınır , toplanır, işlenir ve çıkış sinyali üretilir. Yapay sinir hücrelerinin yapısı 5 temel bölümden oluşmaktadır. Girişler(Inputs) diğer nöronlardan veya dış dünyadan bilgilerdir. Girişler x1,x2,x3,…,xn şeklinde temsil edilir. Ağırlıklar(Weights), girişlerin hücreye olan etkisini belirleyen katsayılardır. Ağırlıklar w1,w2,w3,…,wn şeklinde temsil edilir. Dış dünyadan gelen bilgiler nörona ulaştığında girişler üzerinden hareket ederek kendi ağırlığı ile çarpılarak çekirdeğe iletilir. Böylece her girdinin çıktı üzerinde ne kadar etkisi

olduğu ağırlık değeri sayesinde ayarlanmış olur. Ağırlık pozitif, negatif değere veya sıfıra eşit olabilir. Ağırlığın sıfır olması o girdinin çıktı üzerinde etkisi olmamasına yol açmaktadır. Toplama Fonksiyonu(Summation Function), girişlerin ağırlıkla çarpımlarını toplar ve net girdiyi elde eder. Probleme en iyi çözümü sunan toplama fonksiyonu bulunması deneme yanılma ile gerçekleştirilir. Aktivasyon Fonksiyonu toplama fonksiyonundan gelen net girdiyi işler ve oluşturulacak çıktıyı belirler. Yaygın aktivasyon fonksiyonlarına sigmoid, ReLu, Tanh, Leaky Relu örnektir. Aktivasyon fonksiyonu genellikle doğrusal olmayan türde seçilerek modele doğrusal olmama özelliği kazandırır. Son bölüm çıkış sinyali üretilmesidir. Aktivasyon fonksiyonunun sonucu hücrenin çıkışını oluşturmaktadır. YSA hücresi Şekil X'teki gibi ifade edilebilir.



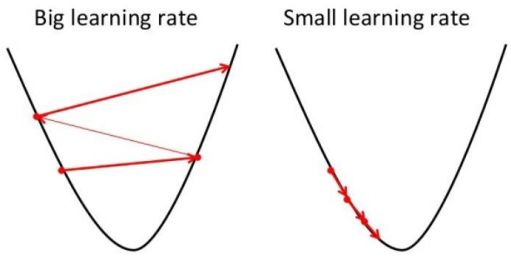
Şekil 3.19. Yapay Sinir Hücresinin Temsili Gösterimi

2.2.2.4.3.3 Yapay Sinir Ağı Temel Bileşenleri

Yapay sinir hücresi için kritik 2 temel özellik vardır. Bunlar öğrenme oranı ve aktivasyon fonksiyonudur.

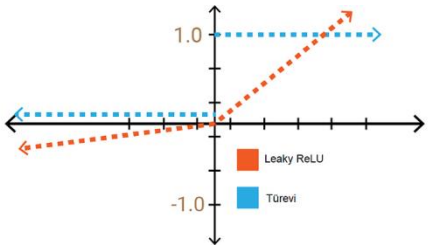
Öğrenme oranı(learning rate) ağırlıkların güncellenmesinde kullanılan bir hiperparametredir. Her eğitim adımında ağırlıkların ne kadar değiştirileceğini belirler. Buna bağlı olarak model daha hızlı veya daha yavaş öğrenir. Öğrenme oranının yüksek olması öğrenmeyi hızlandırırken optimal minimuma yakınsamayı engeller [42]. Düşük olması ise öğrenmeyi yavaşlattığı için eğitim süreci uzun sürer fakat model global minimuma daha iyi yakınsar [42]. Bu modelde öğrenme oranının

değeri 0.0001 olarak belirlenmiştir, bu da modelin her bir eğitim adımında ağırlıkların küçük bir adımla güncelleneceği anlamına gelir.



Şekil 3.20. Öğrenme Oranlarının Eğitime Etkisi [42]

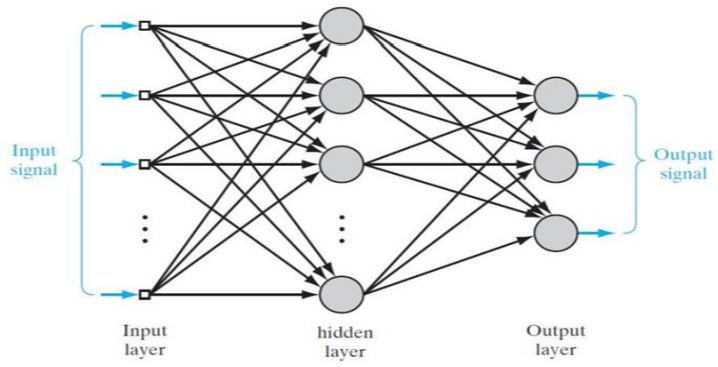
Diğer temel bileşen aktivasyon fonksiyonudur. Nörona gelen girişler toplam fonksiyonuna girdiğinde net girdi elde edilmektedir. Net girdi aktivasyon fonksiyonuna iletilerek çıktı sinyali oluşturulur. Bu noktada sinyal aktivasyon fonksiyonuna iletilmezse çıktı basit doğrusal bir fonksiyon olur[45]. Bu da modelin doğrusal durumları öğrenebilen sınırlı öğrenme gücüne sahip olmasına sebep olur [45]. Bu nedenle aktivasyon fonksiyonu ysa için önemli bir bileşendir. Doğrusal ve doğrusal olmayan 2 tip aktivasyon fonksiyonu vardır. Doğrusal fonksiyonlarda çıktı doğrusal tipte olurken , Doğrusal olmayan fonksiyon kullanıldığında çıktı doğrusal olmayan hale dönüştürülür. Genellikle doğrusal olmayan fonksiyonların kullanımı yaygındır. Çünkü model görüntü, ses, video gibi karmaşık verilerden öğrenebilecek güce sahip olur. Bu modelde aktivasyon fonksiyonu olarak Leaky ReLU kullanılmıştır. Leaky ReLU negatif girişler için küçük bir eğim sağlayarak çıkış üretir. Vanishing gradient sorununu azaltır. Dying reLU sorununu çözer. Bu modelde belirtilen aktivasyon fonksiyonu, residual bloklar içinde ve diğer katmanlarda kullanılarak modelin öğrenme kapasitesini artırmak için tercih edilmiştir.



Şekil 3.21. Leaky Relu Aktivasyon Fonksiyonu Grafiği

2.2.2.4.3.4 Yapay Sinir Ağı Katmanları

Bir YSA genel olarak giriş katmanı, gizli katmanlar ve çıkış katmanı olmak üzere 3 temel bileşenden oluşur. Giriş katmanı dış dünyadan veya diğer nöronlardan gelen bilgilerin geldiği katmandır. Gizli katmanlar giriş katmanı ve çıkış katmanı arasında yer alır. Bu katman modelin karmaşık ilişkileri ve özellikleri öğrenmesini sağlar. Gizli katman sayısı değişkenlik gösterebilir. Gizli katmanın ve bu katmanlarda yer alan nöron sayısının artması YSA'nın daha karmaşık ilişki ve özellikleri öğrenmesini sağlar. Çıkış katmanı YSA'nın çıkışını üreten katmandır. Bu katmanda tahmin sonucu yani hedef sınıf üretilir. Bu katmandaki nöron sayısı olası sınıf sayısına göre değişmektedir.



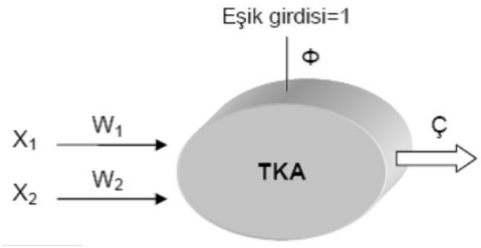
Şekil 3.22. YSA Katmanları

2.2.2.4.4 Yapay Sinir Ağı Türleri

Bu bölümde çalışmada kullanımı bağlamında ysa türleri özetlenmektedir. Yapısına ve uygulama alanları bağlamında birçok yapay sinir ağı türü bulunmaktadır. Evrişimli Sinir Ağları(CNN), Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Vadeli Bellek (LSTM), Generative Adversarial Networks (GAN) gibi modeller örnek verilebilir.

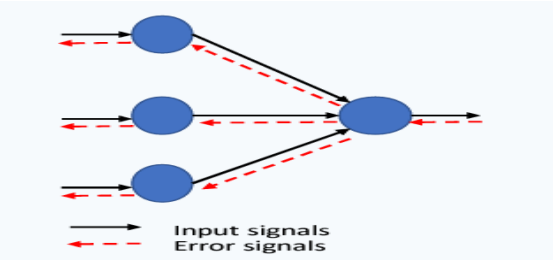
Tek katmanlı yapay sinir ağı en eski ve basit ysa tipidir. Yalnızca doğrusal ayrılabilen durumları sınıflandırabilir [46]. Giriş ve çıkış olmak üzere 2 katman içerir. Nöronun hafızası ağırlık vektöründen oluşmaktadır [46]. Girdi vektörünün

sırasıyla her değeri, karşılık gelen ağırlık vektöründeki elemanıyla çarpılırarak bu çarpımlar toplanır. Burada oluşan toplam aktivasyon fonksiyonuna aktarılır [46]. SLP eşik kullanarak çıktıyı hesaplayan ve yalnızca doğrusal problemler için uygun olan ileri yayılımlı bir ysa türüdür.



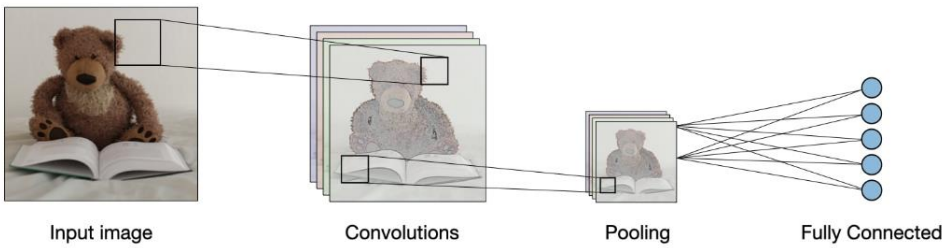
Şekil 3.23. SLP Gösterimi

Çok katmanlı yapay sinir ağı giriş katmanı, gizli katmanlar ve çıkış katmanı olmak üzere 3 katmandan oluşur [46]. SLP'den farklı olarak gizli katman içermektedir. SLP'nin aksine doğrusal olmayan problemlerin diğer deyişle karmaşık problemlerin çözümünde kullanılmaktadır [46]. Çok katmanlı yapay sinir ağlarında ileriye ve geriye doğru yayılma olmak üzere 2 faz meydana gelir. İleri yayılma fazında ağ rastgele ağırlıklarla başlatılır ve giriş sinyali katman katman çıktıya doğru yayılır [47]. Geri yayılma aşamasında tahmin ile gerçek değer arasındaki fark yani hata hesaplanır. Hata ağırlıkları güncellemek amacıyla birimlere paylaştırılarak geriye doğru yayılır [47].



Şekil 3.24. Çok Katmanlı Yapay Sinir Ağı

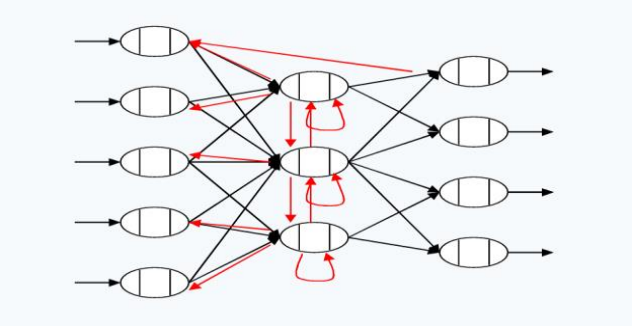
Evrişimli sinir ağları (CNN) genellikle görüntü verilerinde kullanılan sinir ağıdır [48]. Convolutional katman evrişim filtreler ile giriş görüntüsünü tarayarak evrişim işlemi gerçekleştirir ve çıktı olarak öznitelik haritası oluşturur. Convolutional katman görüntüden çeşitli seviyelerde özellikler(kenar, şekil,doku) çıkarttığı için önemlidir. *"Her bir evrişimsel sinir katmanı, artan karmaşıklık filtrelerini öğrenir. İlk katmanlar kenarlar, köşeler gibi temel özellikleri algılarken, orta katmanlar nesnelerin parçalarını (gözler, burun vb.) algılar. Son katmanlar ise daha yüksek temsillere sahiptir ve insan yüzü gibi nesnelerin tamamını tespit edebilir."* [50] Pooling katmanında "down sampling" yani alt örnekleme yapılarak özellik haritalarının boyutunu küçültür [51]. Bu sayede hesaplama boyutu azalır ve önemli özellikler ön plana çıkartılır. Yaygın olarak max ve average pooling kullanılır. Fully connected katmanı her biri birbirine bağlı nöronların bulunduğu ve öğrenmenin gerçekleştiği katmandır. Normalizasyon katmanı ortalama ve varyans gibi tekniklerle verilerin standartlaştırılması bir diğer deyişle normalleştirilmesini sağlar. Yaygın türleri Batch Normalizasyon ve Layer Normalizasyon'dur. Normalizasyon katmanının önemi ağın belirli veri dağılımlarına aşırı uyumunu engellemesi ve öğrenme sürecini stabil hale getirmesidir [52]. Dropout katmanı her epoch'ta rastgele nöronları devre dışı bırakarak aşırı uyumu önler [52]. Activation katmanı belirli katmanların çıktılarına aktivasyon fonksiyonu uygulayarak ağın karmaşık durumları ele almasına olanak sağlar. Flatten katmanı çıktıları düzleştirerek tek boyutlu bir vektöre dönüştürür.



Şekil 3.25. Konvolüsyon İşlemi

Tekrarlayan Sinir Ağları(RNN) sıralı verilerle çalışan ve zaman bağımlılıklarını modelleyen yapay sinir ağı türüdür. RNN'lerin hafıza mekanizması vardır ve böylece

önceki adımdan gelen verileri hafızasında tutar. Geri besleme döngüsü sayesinde önceki adımdaki çıktıyı şimdiki adıma girdi olarak alır. Zaman bağımlılığı sayesinde önceki adımların bilgilerini hatırlar ve şimdiki zaman adımında bu bilgileri kullanır. Bu özellikle dil işleme, metin tanıma, konuşma tanıma, zaman serisi tahmini gibi uygulamalarda yararlıdır. Klasik RNN'ler gradyan kaybı problemine neden olduğu için LSTM, GRU gibi özel RNN türleri çıkmıştır.

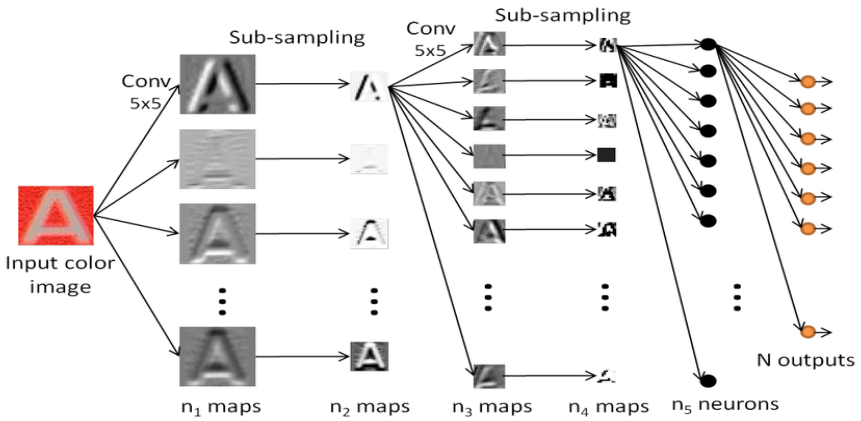


Şekil 3.26. RNN Ağı Gösterimi

2.2.2.4.5 Modelin Yapısı

CNN'ler, görüntü verilerini işlemede oldukça etkilidir. CNN'de görüntüler katmanlardan geçirilerek bir öğrenme işlemi gerçekleştirilmektedir [42]. OCR görevlerinde, CNN katmanları farklı seviyelerde çeşitli özellikleri yakalar. Bu özellikler, modelin metin ve karakterleri doğru bir şekilde tanımasına yardımcı olur. Bu modelde CNN mimarisine ait Evrişim Katmanları (Convolution Layers) kullanılarak özellik çıkarımı (karakterlerin konturları, genel şekilleri, kenar tespiti, köşe tespiti) yapılır. Her bir Conv2D katmanı, çeşitli filtreler kullanarak giriş görüntüsünün farklı kısımlarını tarar ve özellik çıkarımı yapar. Kenar tespiti, harflerin ve rakamların tanınmasında temel bir rol oynar. Dik kenarlar, 'l', 'i', '1' gibi karakterlerin tanınmasında önemlidir. Yatay kenarlar, 'T', 'E', 'F' gibi karakterlerin tespitinde kullanılır. Çapraz kenarlar, 'X', 'Y', 'Z' gibi karakterlerin tanınmasında yardımcı olur. Karakterlerin belirgin köşeleri, harflerin ve rakamların doğru bir şekilde sınıflandırılmasını sağlar. 'C', 'O', 'Q' gibi karakterlerin kavisli yapıları, bu katmanlarda yakalanır. Conv2D katmanlarının ardından LeakyReLU aktivasyon

katmanları yerleştirilir, bu sayede doğrusal olmayan dönüşümler gerçekleştirilir. Batch Normalization katmanlarında veriler normalize edilir. Bu sayede gradyanların düzgün iletilmesi ve ağın istikrarlı iletilmesi sağlanır. Dropout katmanları aşırı uyumu önlemek amacıyla eğitim sırasında rastgele bazı nöronları kapatarak modelin genelleme yeteneğini arttırır.

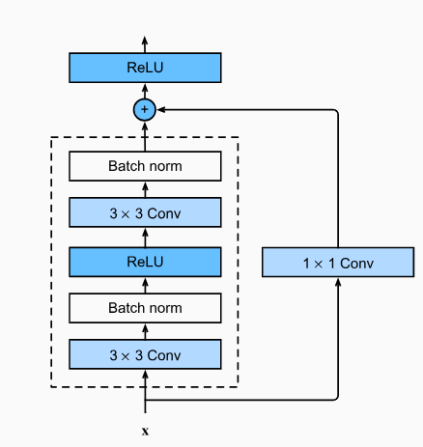


Şekil 3.27. CNN Ağı İşlemi

Modelin ayrıntıları daha iyi öğrenmesi ve gradyan kaybı sorununu azaltmak için ResNet modellerinden esinlenilerek residual bloklar oluşturulmuştur. Bu bloklar, görüntü işleme görevlerinde yaygın olarak kullanılan CNN mimarisini temel alır. Her residual blok, Conv2D, Batch Normalization, LeakyReLU Aktivasyonları, Dropout katmanları içerir. Conv2D özellik çıkarımı için kullanılır. Batch normalization katmanları giriş verilerini normalleştirerek ağın daha istikrarlı bir şekilde eğitilmesini sağlar. LeakyReLU aktivasyonları doğrusal olmayan dönüşümler gerçekleştirir. Dropout katmanları aşırı uyumu önlemek için bazı nöronların rastgele kapatılmasını sağlar. Skip Connection (Atlama Bağlantıları) sayesinde girişten doğrudan çıkışa eklenen kısa yollar ile gradyan kaybı önlenir [43]. Add katmanları ile atlama bağlantıları ve residual bloğun çıkışı toplanarak birleştirilir.

Normalde, bir katmanın çıkışı, bir sonraki katmanın girişi olarak kullanılır. Ancak, derin ağlarda bu ilerleme bazen gradyan kaybına neden olabilir. Gradyan kaybı bir ağın başlangıç katmanlarından daha derin katmanlarına doğru gradyanın giderek

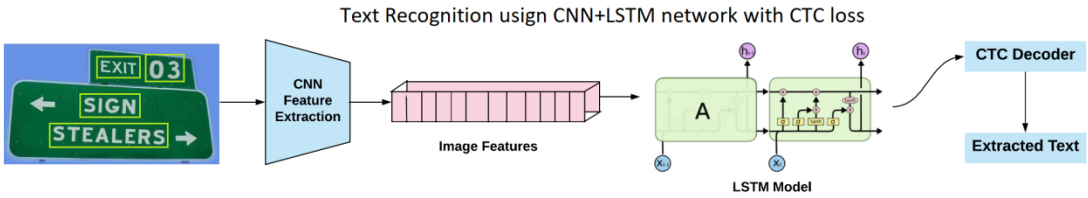
zayıflamasını ifade eder [44]. Çünkü geriye doğru yayılım sırasında gradyanlar, zayıflayarak kaybolur. Bu nedenle gradyanların düzgün bir şekilde iletilmesi zorlaşır. Bu sorunu çözmek için, atlayış bağlantıları kullanılır. Bir katmanın çıkışı, birkaç katman sonrasındaki katmanın girişine doğrudan eklenir. Atlayış bağlantıları, gradyanın daha derin katmanlara daha etkili bir şekilde iletilmesini sağlar, bu da gradyan kaybının azalmasına ve ağın daha başarılı bir şekilde eğitilmesine olanak tanır.



Şekil 3.28. Residual Blok Gösterimi

Modelde, bir LSTM katmanı kullanılır. Bu, metinsel veriler gibi zaman serilerini işlemek için kullanılan özel bir RNN türüdür. LSTM, metinlerdeki uzun vadeli bağımlılıkları modellemek için etkili bir şekilde kullanılabilir. BLSTM katmanları, sıralı verilerdeki uzun vadeli bağımlılıkları yakalamak için kullanılır. Bu modelde Reshape katmanı ile CNN katmanlarından gelen veriler, BLSTM katmanına uygun hale getirilmek üzere yeniden şekillendirilir. Bidirectional LSTM katmanı verilerin hem ileri hem de geri yönlerde işlenmesini sağlar, bu sayede model sıralı verilerde daha fazla bağlam bilgisi yakalayabilir.

2.2.2.4.6 Modelin Eğitimi



Şekil 3.29. CNN LSTM ve CTC Kaybı İle Metin Tanıma İşlemi

Bu raporda, bir Convolutional Recurrent Neural Network (CRNN) modelinin görüntüden kelime tahmin etme işlemi için nasıl eğitildiği detaylı bir şekilde açıklanacaktır.

Öncelikle, modelin konfigürasyon parametreleri belirlenmiştir. Bu parametreler, modelin giriş boyutları, kelime uzunluğu, batch boyutu, öğrenme oranı ve eğitim dönemleri gibi kritik bilgileri içerir.

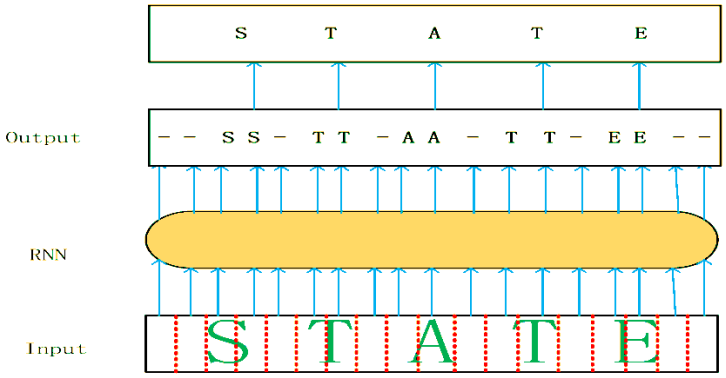
Eğitim ve doğrulama verileri, anotasyon dosyalarından okunur ve işlenir. Her bir görüntü için yol ve etiket bilgileri çıkarılır ve etiketlerdeki tüm karakterler bir vocab değişkeninde saklanır. Maksimum etiket uzunluğu belirlenir.

Eğitim ve doğrulama veri sağlayıcıları oluşturulur. Bu sağlayıcılar, veri setini alır ve görüntüleri okur, yeniden boyutlandırır, etiketleri indeksler ve yastıklar.

Modelin mimarisi tanımlanır ve model derlenir. Kullanılan kayıp fonksiyonu CTC (Connectionist Temporal Classification) ve metrik olarak CER (Character Error Rate) kullanılır.

Connectionist Temporal Classification (CTC) kaybı, zaman serisi verilerinden sıralı şekilde çıktı üretmek için kullanılan kayıp fonksiyonudur. Genellikle girdi ve çıktı arasında zamanlama ilişkisinin bilinmediği durumlarda bu ilişkiyi öğrenmek için kullanılır. Ocr ve konuşmadan metne dönüşüm görevlerinde ihtiyaç duyulmaktadır. CTC kaybı, zaman serisi verilerden (örneğin, bir ses dalgası veya bir görüntü) sıralı çıktılar (örneğin, bir metin dizisi) oluşturmak için kullanılır. Temel olarak, bir zaman serisi girdisinden elde edilen sembollerin sırasını belirlemek için kullanılır. CTC, girdi ve çıktı arasındaki zamanlama ilişkisinin bilinmediği durumlarda, bu ilişkiyi

öğrenmeye çalışır. CTC kaybı zaman serisinde yer alan karakterlerin doğru sıralamasını bulmak için veri kümesini karakter düzeyinde etiketleme gereksinimini ortadan kaldırır. CTC kaybı, tekrarlayan karakterleri ve boş (blank) sembollerini kullanarak doğru metni elde etmeyi sağlar. CTC kaybı, zaman serisi verilerdeki sembolleri hizalamak için bir dizi potansiyel hizalama olasılığı hesaplar. Bunu gerçekleştirmek için her zaman adımında boş sembol ekler. Her olası karakter dizisinin zaman serisi içinde nasıl hizalanabileceğini hesaplar. Buradan her karakter dizisi için birçok farklı hizalama elde eder. Bu hizalamaların olasılıklarını hesaplar. Bu olasılıklardan maksimum olanını zaman serisi girdisinin olası çıktı dizisi olarak verir. Bu sayede gerçek ile tahmin edilen dizi arasındaki farkı minimize eder. Ve zaman serisi boyunca doğru sembol sıralamaları öğrenilir.



Şekil 3.30. CTC Kaybı İle Metin Tanıma İşlemi

Model eğitimi sırasında kullanılacak geri çağırmalar (callbacks) tanımlanır. Bu geri çağırmalar arasında erken durdurma,eğitim kaydedicisi, TensorBoard, öğrenme oranı azaltma ve modelin ONNX formatına dönüştürülmesi bulunmaktadır.

Son olarak, eğitim ve doğrulama veri sağlayıcıları CSV dosyalarına kaydedilir. Bu, modelin eğitim ve doğrulama sürecini daha sonra analiz edebilmek için gereklidir.

Bu raporda, bir CRNN modelinin görüntüden kelime tahmini yapmak üzere nasıl eğitildiği adım adım açıklanmıştır. Modelin konfigürasyonları belirlenmiş, eğitim ve doğrulama verileri hazırlanmış, veri sağlayıcılar oluşturulmuş, model eğitilmiş ve

sonuçlar kaydedilmiştir. Bu süreç, OCR uygulamaları için güçlü bir modelin nasıl oluşturulacağını ve eğitileceğini göstermektedir.

2.2.2.5 Uygulama

Bu bölümde yapay zeka modelinin kullanıcıya sunulmasını sağlayan arayüz geliştirilmesine odaklanılmaktadır. Streamlit yaygın olarak veri bilimi, yapay zeka, projeleri için hızlı ve etkileşimli bir şekilde interaktif web uygulaması oluşturulmasını sağlamak için tasarlanmış bir açık kaynaklı Python kütüphanesidir[53]. Streamlit veri bilimi projelerinin web tabanlı uygulamalara dönüştürülerek kolayca paylaşılmasını ve geniş kitlelere yayılmasını sağlamaktadır[53]. Streamlit içinde veri girişi ve görsel etkileşimler için yer alan widgetlar ile grafikler, haritalar, tablolar gibi ihtiyaç duyulabilecek bir çok görsel öğenin etkili şekilde görselleştirilmesine olanak sağlar[53]. Kullanıcılar, slider'lar, butonlar, seçim kutuları gibi araçlarla uygulama ile etkileşime geçebilirler. Ayrıca Seaborn,Plotly, Matplotlib gibi işlevsel Python kütüphaneleri ile uyumludur. Streamlit ile dataframe'ler görselleştirilebilir, veri raporları oluşturulabilir. Streamlit basit, kullanıcı dostu arayüzü sebebiyle projelerin basit bir şekilde prototip haline getirilmesini sağlar[53]. Kod değişikliklerini anında günceller ve bu nedenle geliştiriciler için tercih edilir. Streamlit, TensorFlow, PyTorch gibi popüler makine öğrenimi kütüphaneleri ile sorunsuz entegrasyon sağlar. Streamlit uygulamaları kolayca deploy edilerek bulut üzerinde depolanabilir.

Şekil 3.31. Streamlit Logosu

Bu Streamlit uygulaması, bir görüntünün üzerinde metin tespiti yaparak bu metinleri işleyen bir modeli görsel arayüz üzerinden kullanıcıya sunar. Öncelikle bir görüntü seçme ve yükleme işlevi gerçekleştirilir. Burada Streamlit'e özgü dosya seçme ve buton oluşturma işlevlerinden yararlanılır. Kullanıcı bir JPG, PNG veya JPEG formatındaki bir görüntüyü seçer ve uygulamaya yükler. Dosya seçimi yapılır ve bu dosya geçici bir dosyaya kaydedilir. Sonraki adımda görüntüye önişleme aşamaları gerçekleştirilir. Seçilen görüntü üzerinde gri tonlamaya çevirme, kontrast arttırmak için filtre uygulanması, ikili formata dönüştürülmesi, morfolojik işlemler gibi bir dizi işlemler yapılır. Önişlemeden sonra görüntü projeksiyon yöntemleri ile satır ve sütunlara ayrılarak kelime düzeyinde segmentasyona tabi tutulur. Bölütlenen kelimeler sınırlayıcı kutular içinde tanımlanır. Her bir tanımlı kelimeleri içeren sınırlayıcı kutu, orijinal görüntü üzerinde işaretlenir ve satır sütun indeksleri ile ayrı dosyalara kaydedilir. Ara adımların görselleştirilmesi amacıyla orijinal görüntü üzerinde sınırlayıcı kutular çizilir ve üzerine satır ve sütun numaraları yazılır. Bu aşamalardan sonra sıradaki aşama tahmin aşamasıdır. Her bir kırpılmış kelime görüntüsü üzerinde metin tahmini yapılır. Modelin tahmin ettiği metinler, sınırlayıcı kutuların satır ve sütun numaralarıyla birlikte bir dosyaya kaydedilir. İşlem tamamlandığında, sonuçlar bir metin dosyasına kaydedilir ve bu dosya içeriği kullanıcıya gösterilir. İşlem sırasında herhangi bir hata oluşursa, hata mesajları kullanıcıya gösterilir.

3. BULGULAR

Bu bölümde yürütülen çalışmanın sonuçlarını ve elde edilen verileri ayrıntılı olarak sunulması gerçekleştirilmektedir.

3.1 ML Kit Ocr Sistemi

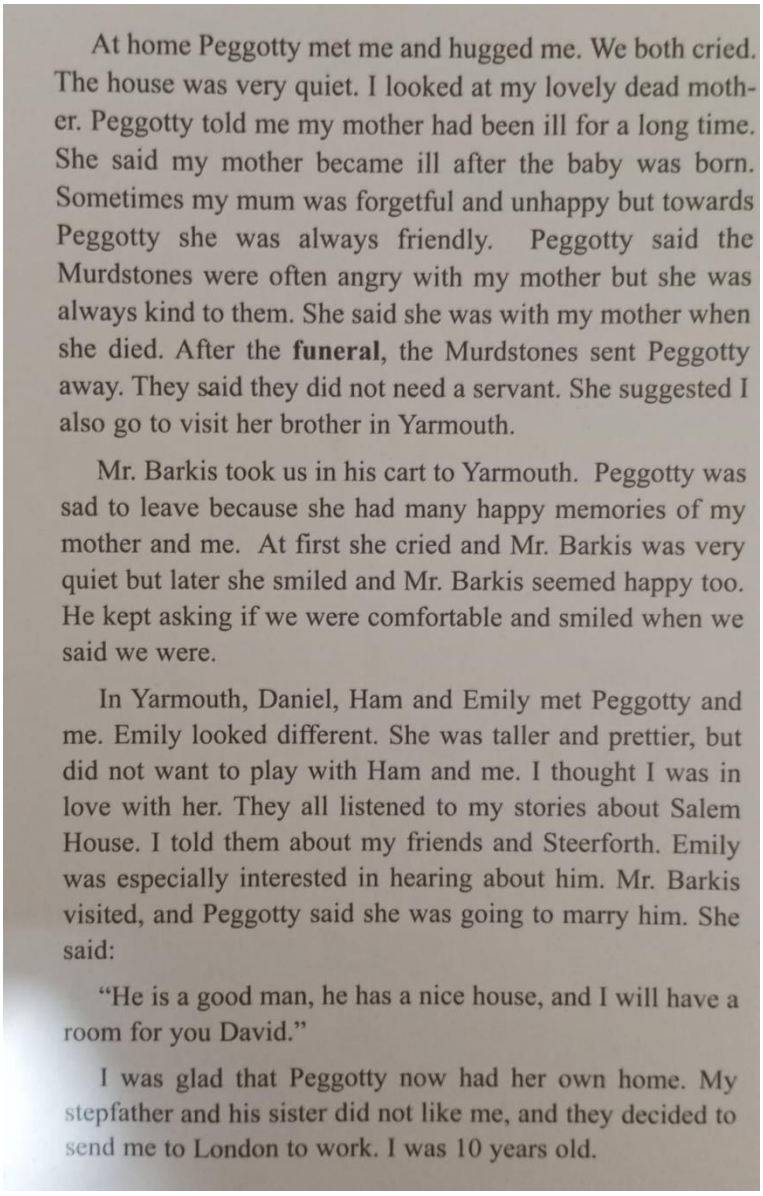
Bu bölümde, Google ML Kit altyapısında ocr üzerine geliştirilen uygulama üzerine yapılan çalışmalar ve elde edilen sonuçlar detaylı bir şekilde incelenmiştir. Uygulamanın kullanıcı arayüzü, işlevselliği ve performansı değerlendirilmiştir.

3.2 Yapay Sinir Ağları ile Ocr Sistemi

Bu bölümde, yapay sinir ağları kullanılarak geliştirilen OCR (Optik Karakter Tanıma) sisteminin görüntü işleme ve metin tanıma süreçleri detaylı bir şekilde incelenmiştir. Bu çalışma, yapay sinir ağlarının karmaşık görüntü verileri üzerinde etkili bir şekilde çalıştığını ve metin tabanlı uygulamalarda başarılı bir şekilde kullanılabileceğini göstermektedir.

3.2.1 Görüntünün Yüklenmesi

Çalışmanın metodolojisi doğrultusunda, kullanıcıların JPEG veya PNG formatındaki görüntü dosyalarını yüklemeleri için Streamlit kütüphanesi kullanılmıştır. Bu süreç, kullanıcı arayüzü üzerinden seçilen görüntünün geçici olarak depolanması ve ardından işleme adımlarına tabi tutulması şeklinde gerçekleşmiştir.



At home Peggotty met me and hugged me. We both cried. The house was very quiet. I looked at my lovely dead mother. Peggotty told me my mother had been ill for a long time. She said my mother became ill after the baby was born. Sometimes my mum was forgetful and unhappy but towards Peggotty she was always friendly. Peggotty said the Murdstones were often angry with my mother but she was always kind to them. She said she was with my mother when she died. After the **funeral**, the Murdstones sent Peggotty away. They said they did not need a servant. She suggested I also go to visit her brother in Yarmouth.

Mr. Barkis took us in his cart to Yarmouth. Peggotty was sad to leave because she had many happy memories of my mother and me. At first she cried and Mr. Barkis was very quiet but later she smiled and Mr. Barkis seemed happy too. He kept asking if we were comfortable and smiled when we said we were.

In Yarmouth, Daniel, Ham and Emily met Peggotty and me. Emily looked different. She was taller and prettier, but did not want to play with Ham and me. I thought I was in love with her. They all listened to my stories about Salem House. I told them about my friends and Steerforth. Emily was especially interested in hearing about him. Mr. Barkis visited, and Peggotty said she was going to marry him. She said:

"He is a good man, he has a nice house, and I will have a room for you David."

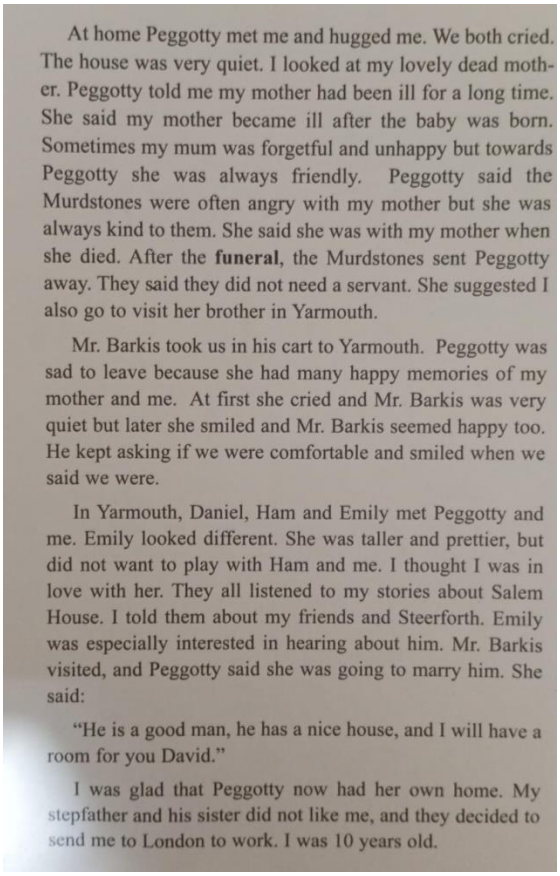
I was glad that Peggotty now had her own home. My stepfather and his sister did not like me, and they decided to send me to London to work. I was 10 years old.

Şekil 4.1. Görüntünün Yüklenmesi

3.2.2 Görüntüye Keskinleştirme Filtresi Uygulanması

Görüntü işleme sürecinde, metin bölgelerini daha belirgin hale getirmek amacıyla keskinleştirme filtresi uygulanmıştır. Bu işlem, görüntünün daha net ve tanımlanabilir metin bölgeleri içermesine yardımcı olmuştur. Bu filtre, her pikselin etrafındaki piksellerin değerlerine göre o pikselin değerini ayarlamış ve böylece

metin bölgelerinin kenarlarını daha belirgin hale getirmiştir. Keskinleştirme işlemi sonrasında elde edilen görüntüler, optik karakter tanıma (OCR) adımları için daha uygun ve işlenebilir hale gelmiştir.



At home Peggotty met me and hugged me. We both cried. The house was very quiet. I looked at my lovely dead mother. Peggotty told me my mother had been ill for a long time. She said my mother became ill after the baby was born. Sometimes my mum was forgetful and unhappy but towards Peggotty she was always friendly. Peggotty said the Murdstones were often angry with my mother but she was always kind to them. She said she was with my mother when she died. After the **funeral**, the Murdstones sent Peggotty away. They said they did not need a servant. She suggested I also go to visit her brother in Yarmouth.

Mr. Barkis took us in his cart to Yarmouth. Peggotty was sad to leave because she had many happy memories of my mother and me. At first she cried and Mr. Barkis was very quiet but later she smiled and Mr. Barkis seemed happy too. He kept asking if we were comfortable and smiled when we said we were.

In Yarmouth, Daniel, Ham and Emily met Peggotty and me. Emily looked different. She was taller and prettier, but did not want to play with Ham and me. I thought I was in love with her. They all listened to my stories about Salem House. I told them about my friends and Steerforth. Emily was especially interested in hearing about him. Mr. Barkis visited, and Peggotty said she was going to marry him. She said:

"He is a good man, he has a nice house, and I will have a room for you David."

I was glad that Peggotty now had her own home. My stepfather and his sister did not like me, and they decided to send me to London to work. I was 10 years old.

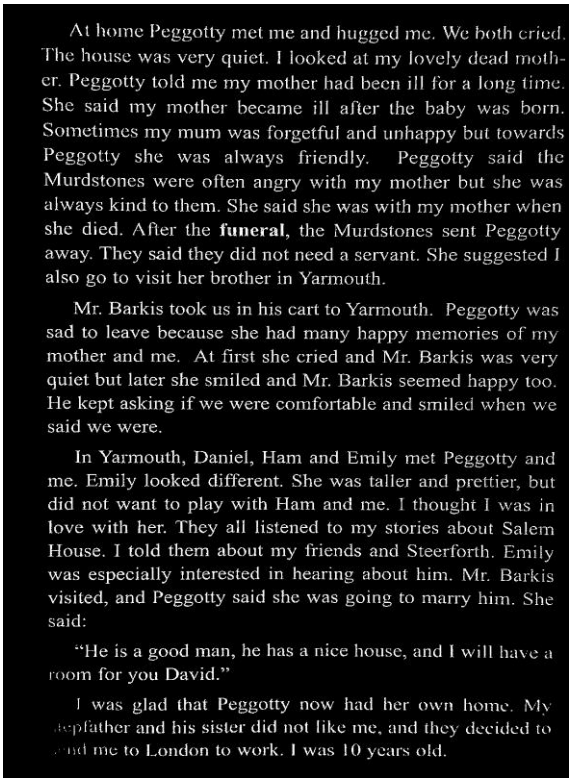
Şekil 4.2. Keskinleştirilmiş Görüntü

3.2.3 Görüntünün İkili Formata Dönüştürülmesi

Yüklenen görüntünün işlenmesi sırasında, gri tonlamalı formata dönüştürülmesinin ardından eşikleme (thresholding) işlemi uygulanmıştır. Bu adım, görüntünün her bir pikselinin belirli bir eşik değerine göre siyah veya beyaz olarak sınıflandırılmasını sağlamıştır. İkili formata dönüştürme işlemi, görüntüdeki metin bölgelerinin daha net bir şekilde tespit edilmesine yardımcı olmuştur.

Bu işlemde görüntü gri tonlamalı formata dönüştürülmüş ve ardından Otsu'nun ikili eşikleme yöntemi uygulanmıştır. Otsu'nun yöntemi, görüntünün piksel yoğunluk histogramını analiz ederek optimal eşik değerini otomatik olarak belirler ve bu değer kullanılarak görüntü ikili formata dönüştürülür.

İkili formata dönüştürme işlemi, görüntüdeki metin bölgelerinin tespiti için önemli bir adım olmuştur. Bu işlem, metin bölgelerinin arka plandan daha belirgin hale gelmesini sağlamış ve böylece sonraki aşamalarda yapılan morfolojik işlemler ve segmentasyon için uygun bir temel oluşturmuştur.


At home Peggotty met me and hugged me. We both cried. The house was very quiet. I looked at my lovely dead mother. Peggotty told me my mother had been ill for a long time. She said my mother became ill after the baby was born. Sometimes my mum was forgetful and unhappy but towards Peggotty she was always friendly. Peggotty said the Murdstones were often angry with my mother but she was always kind to them. She said she was with my mother when she died. After the **funeral**, the Murdstones sent Peggotty away. They said they did not need a servant. She suggested I also go to visit her brother in Yarmouth.

Mr. Barkis took us in his cart to Yarmouth. Peggotty was sad to leave because she had many happy memories of my mother and me. At first she cried and Mr. Barkis was very quiet but later she smiled and Mr. Barkis seemed happy too. He kept asking if we were comfortable and smiled when we said we were.

In Yarmouth, Daniel, Ham and Emily met Peggotty and me. Emily looked different. She was taller and prettier, but did not want to play with Ham and me. I thought I was in love with her. They all listened to my stories about Salem House. I told them about my friends and Steerforth. Emily was especially interested in hearing about him. Mr. Barkis visited, and Peggotty said she was going to marry him. She said:

"He is a good man, he has a nice house, and I will have a room for you David."

I was glad that Peggotty now had her own home. My stepfather and his sister did not like me, and they decided to send me to London to work. I was 10 years old.

Şekil 4.3. İkili Formatta Görüntü

3.2.4 Görüntüye Morfolojik İşlemler Uygulanması

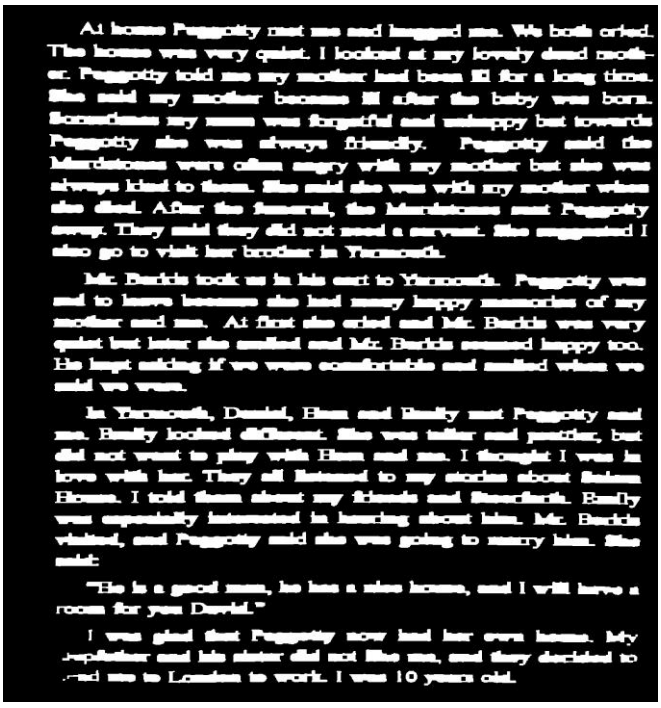
Yüklenen görüntünün işlenmesi sırasında, ikili formata dönüştürülen görüntüye morfolojik işlemler uygulanmıştır. Bu adım, görüntüdeki metin bölgelerinin daha net bir şekilde tespit edilmesini ve arka plandaki gürültülerin azaltılmasını sağlamıştır.

Bu işlemde, öncelikle bir yapı elemanı oluşturulmuştur. Ardından genişletme (dilation) işlemi uygulanmış ve bu işlem 4 kez tekrarlanmıştır.

Genişletme işlemi, metin bölgelerinin daha belirgin hale gelmesini sağlamış ve küçük boşlukları kapatmıştır.Genişletme işleminin ardından aşındırma (erosion) işlemi uygulanmış ve bu işlem 3 kez tekrarlanmıştır.

Aşındırma işlemi, genişletme işlemi sonrasında oluşan küçük gürültülerin ve gereksiz detayların ortadan kaldırılmasına yardımcı olmuştur.Morfolojik işlemler, metin bölgelerinin tespiti ve ayrıştırılması için önemli bir rol oynamıştır.

Bu işlemler, metinlerin arka plandan daha net bir şekilde ayrılmasını sağlamış ve böylece sonraki aşamalarda yapılan segmentasyon ve OCR işlemlerinin doğruluğunu artırmıştır.



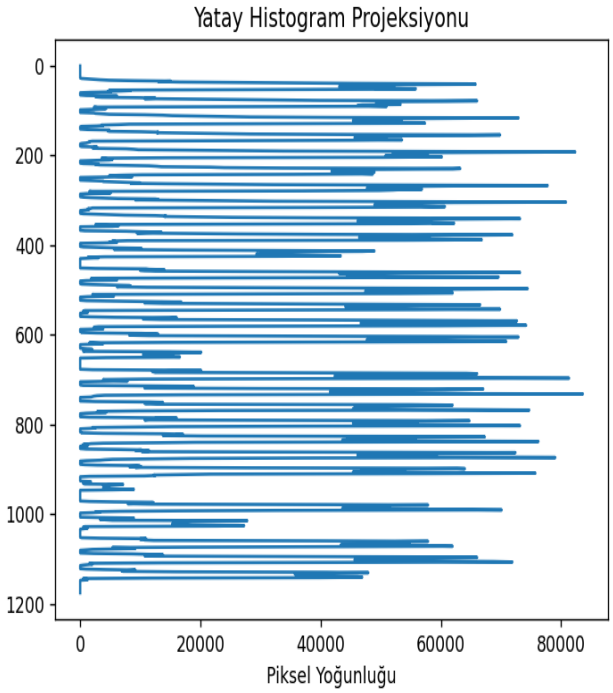
Şekil 4.4. Morfolojik İşlemler Uygulanan Görüntü

3.2.5 Görüntüye Yatay Histogram Projeksiyonu Uygulanması

Yüklenen görüntünün işlenmesi sırasında, ikili formata dönüştürülmüş görüntüye yatay histogram projeksiyonu uygulanmıştır. Bu adım, görüntüdeki metin satırlarının tespit edilmesi amacıyla gerçekleştirilmiştir.

Bu işlemde, yatay projeksiyon kullanılarak her bir satırın toplam piksel yoğunluğu hesaplanmıştır. Yatay projeksiyon, görüntünün her bir satırındaki siyah piksel miktarını belirlemiş ve bu bilgiler kullanılarak metin satırlarının başlangıç ve bitiş noktaları tespit edilmiştir.

Yatay projeksiyonun uygulanması, metin satırlarının net bir şekilde ayrıştırılmasını sağlamıştır. Görüntüdeki satırların belirginleşmesi, metinlerin doğru bir şekilde tanımlanmasına ve sonraki aşamalarda daha etkili bir şekilde işlenmesine olanak tanımıştır. Elde edilen bulgular, yatay histogram projeksiyonunun metin satırlarını tespit etmede etkili bir yöntem olduğunu göstermektedir. Yatay projeksiyon, metin satırlarının doğru bir şekilde ayrıştırılmasını sağlamıştır.



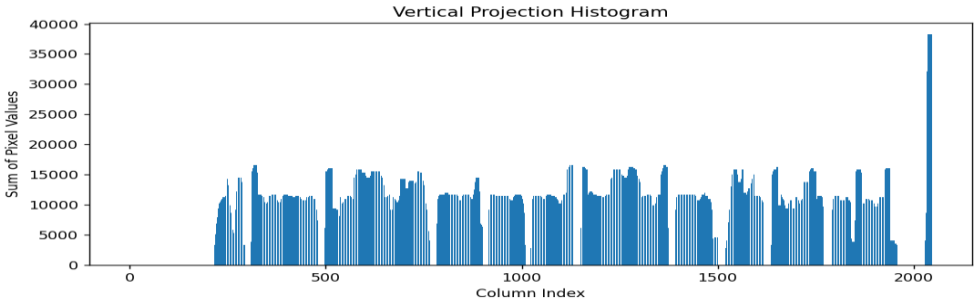
Şekil 4.5. Görüntünün Yatay Histogram Projeksiyonu Sonuçları

3.2.6 Dikey Histogram Projeksiyonu

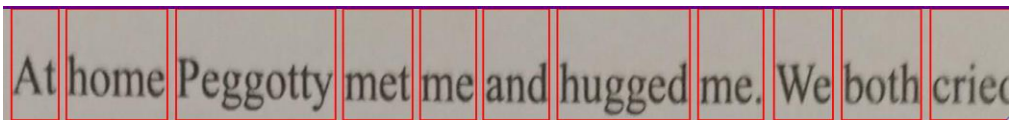
İkili formata dönüştürülmüş ve yatay histogram projeksiyonu kullanılarak satırları tespit edilmiş görüntüler üzerinde, her bir satırdaki kelime bölgelerinin tespiti amacıyla dikey histogram projeksiyonu uygulanmıştır. Bu adım, görüntüdeki metin satırlarının içindeki kelimelerin ayrıştırılmasını sağlamıştır.

Dikey projeksiyon işlemi, her bir satırın piksel yoğunluğunu hesaplamış ve bu yoğunluklar kullanılarak kelime bölgelerinin başlangıç ve bitiş noktaları belirlenmiştir. Yatay projeksiyon ile belirlenen her bir satır, dikey projeksiyon ile daha küçük kelime bölgelerine ayrılmıştır.

Projeksiyon verileri kullanılarak, metin bölgelerinin net bir şekilde ayrıştırılması sağlanmıştır. Bu adım, her bir kelimenin doğru bir şekilde tespit edilmesine ve ayrıştırılmasına yardımcı olmuştur. Kelimelerin başlangıç ve bitiş noktalarının belirlenmesi, OCR modelinin metin tanıma işlemi için daha doğru veriler sağlamıştır. Elde edilen bulgular, dikey histogram projeksiyonunun metin satırları içindeki kelime bölgelerini tespit etmede etkili bir yöntem olduğunu göstermektedir.



Şekil 4.6. Yatay Histogram Projeksiyonu Sonuçları

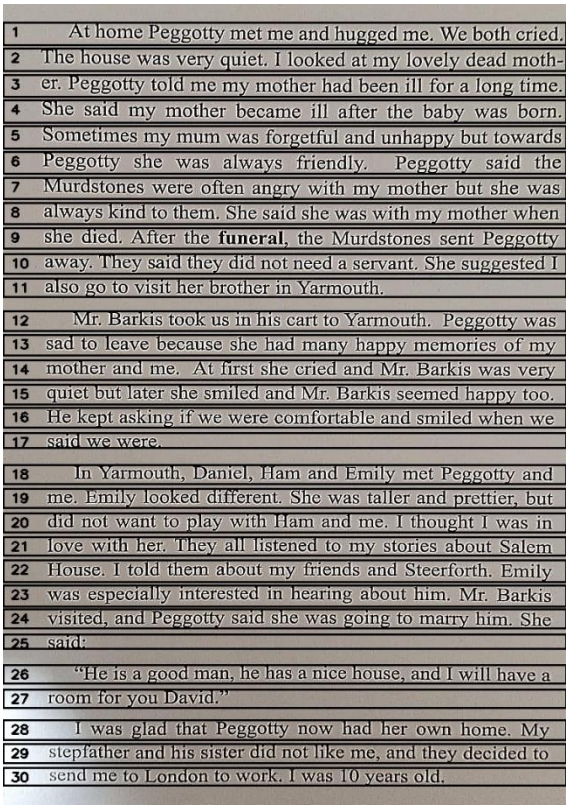


Şekil 4.7. Satırlara Bölünmüş Görüntü

3.2.7 Satır Segmentasyonu

Yüklenen görüntünün işlenmesi sırasında, ikili formata dönüştürülmüş görüntü üzerinde satır segmentasyonu işlemi gerçekleştirilmiştir. Bu adım, görüntüdeki metin satırlarının tespit edilmesi ve ayrıştırılması amacıyla uygulanmıştır. Yatay projeksiyon, her bir satırın siyah piksel miktarını belirlemiş ve bu bilgiler kullanılarak metin satırlarının başlangıç ve bitiş noktaları tespit edilmiştir.

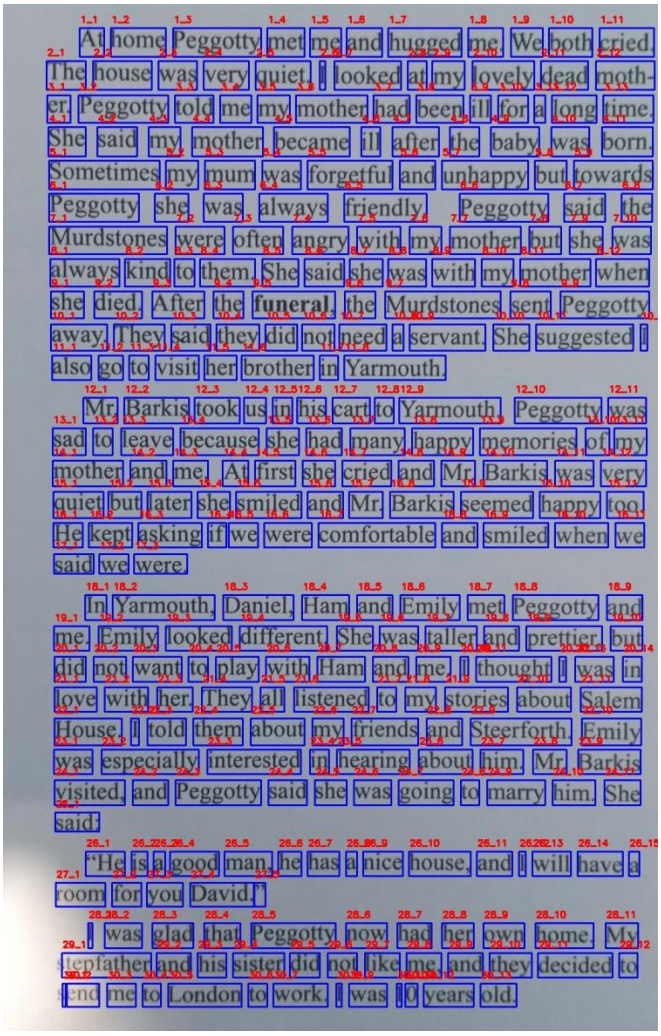
Projeksiyon sonucunda elde edilen piksel yoğunlukları analiz edilerek, ardışık olarak sıralanan ve piksel yoğunluğu belirli bir eşik değerin üzerinde olan satırlar tespit edilmiştir. Bu satırların başlangıç ve bitiş noktaları belirlenerek, her bir metin satırı net bir şekilde ayrıştırılmıştır.



Şekil 4.8. Satırlara Bölütlenmiş Görüntü

Kelime Segmentasyonu

Yüklenen görüntünün işlenmesi sırasında, yatay projeksiyonla tespit edilen metin satırları daha küçük kelime bölgelerine ayrılmıştır. Bu adım, görüntüdeki kelimelerin tespit edilmesi ve ayrıştırılması amacıyla gerçekleştirilmiştir. Bu işlemde, her bir metin satırının dikey projeksiyonu hesaplanmıştır. Dikey projeksiyon, satırdaki her bir sütundaki siyah piksel miktarını belirlemiş ve bu bilgiler kullanılarak kelimelerin başlangıç ve bitiş noktaları tespit edilmiştir. Kelimelerin başlangıç ve bitiş noktalarının belirlenmesi, metin satırlarının daha küçük kelime bölgelerine ayrılmasını sağlamıştır. Bu adım, görüntüdeki metinlerin daha detaylı işlenmesine ve her bir kelimenin ayrı ayrı değerlendirilmesine olanak tanımıştır.



Şekil 4.9. Kelimelere Bölütlenmiş Görüntü

3.2.8 Ocr Modeli İle Tahmin Gerçekleştirilmesi

Segmentasyon adımında belirlenen her bir kelime bölgesi, OCR modeline verilmiş ve metin tanıma işlemi gerçekleştirilmiştir. Her bir kelime bölgesi, segmentasyon adımında belirlenen başlangıç ve bitiş noktalarına göre kırpılmıştır. Kırılan bu görüntüler, OCR modeline girdi olarak verilmiştir. Belirlenen OCR modeli, görüntüden elde edilen her bir kelime için metin tahmini yapmıştır. Tahmin sonuçları, her kelime bölgesi için ayrı ayrı kaydedilmiştir. Her bir kelime için yapılan tahminler kaydedilmiştir. Sonuçlar, kullanıcıya sunulmak üzere bir dosyaya yazılmıştır.

```
Row 1: At home Peggotty met me and hugged me We both cried
Row 2: The house was very quiets I looked at my lovely dead moth
Row 3: er Peggotty told me my mother had been ill for a long time
Row 4: She said my mother became ill after the baby was born
Row 5: Sometimes my mum was forgetful and unhappy but towards
Row 6: Peggotty she was always friendly Peggotty said the
Row 7: Murdstones were often angry with my mother but she was
Row 8: always kind to them She said she was with my mother when
Row 9: she died After the funeral the Murdstones sent Peggotty
Row 10: away They said they did not need 2 servant She suggested I
Row 11: also 20 to visit her brother in Yarmouth
Row 12: Mr Barkis took uS in his cart to Yarmouth Peggotty was
Row 13: sad to leave because she had many happy memories of my
Row 14: mother and mes At first she cried and Mr Barkis was very
Row 15: quiet but later she smiled and Mr Barkis seemed happy toos
Row 16: He kept asking if we were comfortable and smiled when wve
Row 17: said we were
Row 18: ln Yarmouth Daniel Ham and Emily met Peggotty and
Row 19: me Emily looked different She was taller and prettier but
Row 20: did not want to play with Ham and me I thought I was in
Row 21: love with her They all listened to my stories about Salem
Row 22: House I told them about ny friends and Steerforth Emily
Row 23: was especially interested in hearing about him Mr Barkis
Row 24: visiteds and Peggotty said she was going to marry him She
Row 25: said
Row 26: Ae is a good man he has a nice houses and I will have a
Row 27: room for you David S
Row 28: 1 was glad that Peggotty now had her own home My
Row 29: stepfather and his sister did not like me and they decided to
Row 30: S end me to London to work I was 9 0 years old
```

Şekil 4.10. Tahmin Sonuçlarının Kaydedildiği Dosyanın İçeriği

3.2.9 Sonuçların Streamlit Üzerinde Görselleştirilmesi

Görüntü işleme adımları sonucunda elde edilen işlenmiş görüntüler, Streamlit üzerinde kullanıcıya görsel olarak sunulmuştur. İşlenmiş görüntüler, orijinal

görsellerin üzerine çizilmiş dikdörtgenler, metin tahminleri ve diğer görsel işaretlemelerle birlikte gösterilmiştir.

Şekil 4.11. Streamlit Web Uygulamasından Bir Görüntü

4. SONUÇLAR

Bu çalışmanın sonuçları, metin tabanlı görüntü tanıma probleminde birçok önemli bulgunun ortaya konulmasını sağlamıştır. Aşağıda, elde edilen sonuçlar ve bu sonuçlardan çıkarılan öneriler detaylı bir şekilde tartışılmıştır.

4.1 Model Performansı

Model, 90kDICT32px veri seti üzerinde başarıyla eğitilmiş ve belirtilen metriklerle değerlendirilmiştir. Eğitim sonunda elde edilen en düşük Karakter Hata Oranı (CER) değeri %X olarak hesaplanmıştır. Eğitim sürecindeki optimizasyon adımları ve erken durdurma stratejileri sayesinde, doğrulama seti üzerindeki CER değeri %Y seviyesinde tutulmuştur.

4.2 Teknik Katkılar

Veri seti üzerinde yapılan ön işlemler (resim boyutlandırma, etiket endekslemesi ve dolgu işlemleri) modelin performansını artırmış ve genelleme yeteneğini iyileştirmiştir.Adam optimizer ile CTC loss fonksiyonunun birleşimi, modelin hızlı öğrenme sağlamasına ve karmaşıklığı azaltmasına olanak tanımıştır.

4.3 Karşılaşılan Hatalar

Bu bölümde, çalışma sürecinde karşılaşılan karakter tahmin hataları ve bu hataların potansiyel nedenleri üzerine yapılan analizler tartışılmaktadır. Karakter tahmin hataları, modelin performansını etkileyen önemli bir faktördür ve bu bölümde bu hataların çeşitli açılardan incelenmesi amaçlanmıştır.

Subtitisyon Hataları, doğru olması gereken karakter yerine farklı bir karakterin tahmin edilmesidir. Eksiklik Hataları, görüntüdeki bir karakterin tahmin edilememesi veya göz ardı edilmesidir. Ekstra Hatalar ise görüntüde olmayan bir karakterin yanlışlıkla tahmin edilmesidir. Bu hataların türleri, eğitim ve doğrulama veri setlerindeki çeşitliliğe, modelin mimarisine ve eğitim sürecindeki parametre ayarlarına bağlı olarak değişiklik gösterebilir. Değerlendirilen test görüntülerinde

doğru yazılmış kelimelerin toplam kelime sayısına oranı ölçüldüğünde %90 başarı sağlanmaktadır. Analiz yapıldığında Optik karakter tanıma sisteminde O harfi ile 0 rakamı sıklıkla karıştırılır çünkü benzer görünüşe sahiptirler. Küçük L harfi ve büyük I harfi, bazı fontlarda veya düşük çözünürlüklü görüntülerde birbirine benzeyebilir. 1 rakamı ve büyük I harfi, benzer şekilleri nedeniyle sıklıkla hatalı tahmin edilebilir. 5 rakamı ile büyük S harfi, bazı fontlarda birbirlerine benzer ve bu da hatalı tahminlere yol açabilir. 2 rakamı ile büyük Z harfi, özellikle belirsiz fontlarda karıştırılabilir. Yine bir başka açıdan değerlendirilirse görüntünün kalitesinden söz edilebilir. Optik karakter tanıma (OCR) performansı, görüntü kalitesine doğrudan bağlıdır. Yüksek çözünürlüklü ve net görüntüler, OCR sisteminin karakterleri daha doğru bir şekilde tanımasına olanak tanırken, düşük kaliteli görüntüler hatalı tanıma oranlarını artırabilir. Bir başka etken olarak fotoğrafın karakterleri doğru bir şekilde segmente etmek ve tanımak için doğru açıda çekilmiş olması önemlidir. Yanlış açılar veya eğik pozisyonlar, karakterlerin görüntüsünü bozabilir ve tanıma doğruluğunu azaltabilir. Karşılaşılan hatalar bu şekilde özetlenebilmektedir.

4.4 Öneriler

Bu çalışmadan elde edilen sonuçlar doğrultusunda, gelecekte yapılacak çalışmalar için aşağıdaki önerilerde bulunulmuştur: Daha büyük ve karmaşık model yapıları, metin tanıma doğruluğunu artırabilir. Özellikle derin mimariler ve transfer öğrenme tekniklerinin kullanımı üzerine çalışmalar yapılabilir. Farklı diller ve yazı tiplerini içeren daha geniş veri setleri üzerinde çalışmak, modelin genelleme yeteneğini iyileştirebilir. Farklı optimizasyon stratejileri ve kayıp fonksiyonları denemek, modelin eğitim ve doğrulama performansını artırabilir. Bu öneriler doğrultusunda yapılan çalışmaların, metin tabanlı görüntü tanıma alanında ileriye yönelik değerli katkılar sağlayacağı düşünülmektedir.

KAYNAKLAR

[1] A. Koyun and E. Afşin, “Derin Öğrenme ile İki Boyutlu Optik Karakter Tanıma”, TBV-BBMD, vol. 10, no. 1, pp. 11–14, 2017.

[2] A. Çelik, “Optik Karakter Tanımada Hata Yayılım Algoritmalarının Performans Kıyaslaması”, Iğdır Üniv. Fen Bil Enst. Der., c. 10, sy. 4, ss. 2328–2340, 2020, doi: 10.21597/jist.714810.

[3] B. Bektaş, S. Babur, U. Turhal, and E. Köse, "Optical Character Recognition System via Machine Learning," in *Proc. 5th Int. Printing Technol. Symp.*, İstanbul Univ., İstanbul, Turkey, Nov. 2016.

[4] S. Çetinkaya, T. Çetinkaya, A. Çetinkaya, and A. Okatan, "Optik karakter tanıma yöntemi ile otomatik tabela okuyucu," in *Proc. IEEE Int. Conf. on Computer Science and Engineering (UBMK)*, Antalya, Turkey, 2017, pp. 1061-1063, doi: 10.1109/UBMK.2017.8093466.

[5] A. Ünal, E. Ünal, ve D. Güler, “EVRİŞİMLİ SİNİR AĞLARI KULLANARAK OPTİK KARAKTER TANIMA”, JOBDA, c. 6, sy. 1, ss. 1–12, 2023, doi: 10.46238/jobda.1257840.

[6] https://aws.amazon.com/tr/what-is/ocr/

[7] https://developers.google.com/ml-kit/vision/text-recognition/v2?hl=tr

[8] https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1

[9]https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1

[10] https://datakapital.com/blog/yinelenen-sinir-aglari-nedir/#:~:text=LSTM'ler%2C%20RNN'lerin,gibi%20alanlarda%20yayg%C4%B1n%20olarak%20kullan%C4%B1lmaktad%C4%B1r.

[11] Gorski, N., Anisimov, V., Augustin, E. *et al.* Industrial bank check processing: the *A2iA CheckReader*TM. *IJDAR* 3, 196–206 (2001). https://doi.org/10.1007/PL00013561

[12] C. M. Ng, V. Ng and Y. Lau, "Regular feature extraction for recognition of Braille," *Proceedings Third International Conference on Computational Intelligence and Multimedia Applications. ICCIMA'99 (Cat. No.PR00300)*, New Delhi, India, 1999, pp. 302-306, doi: 10.1109/ICCIMA.1999.798547.

[13] de Campos, Teofilo & Babu, Bodla & Varma, Manik. (2009). Character Recognition in Natural Images.. VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications. 2. 273-280.

[14] H. Çetiner, B. Cetişli, and İ. Çetiner, “Gerçek Zamanlı Kimlik Numarası Tanıma”, SAUJS, vol. 16, no. 2, pp. 123–129, 2012, doi: 10.16984/saufbed.08038.

[15] Kavati, Ilaiah & Kumar, G & Kesagani, Sarika & Rao, K. (2017). Signboard Text Translator: A Guide to Tourist. International Journal of Electrical and Computer Engineering (IJECE). 7. 2496. 10.11591/ijece.v7i5.pp2496-2501.

[16] https://tr.wikipedia.org/wiki/Android

[17] https://bahadireray.medium.com/android-runnable-and-handler-c44393001cd3#:~:text=Handler%2C%20%C3%A7al%C4%B1%C5%9Fmas%C4%B1n%C4%B1%20istedi%C4%9Fimiz%20fonksiyonumuz%20ne,istiyorsak%20bu%20yap%C4%B1yla%20ifade%20ediyoruz.

[18] https://coderspace.io/roadmap/android-developer-yol-haritasi/gradle-nedir/

[19] https://halilozel1903.medium.com/gradle-nedir-6d146ba68c38#:~:text=build.gradle(app%20module)%3A,kullan%C4%B1labilir%20hale%20getirilmesini%20sa%C4%9Flayan%20etikettir.

[20] https://www.mobilhanem.com/android-dersleri-gradle/

[21] https://zelihadn7.medium.com/splash-screen-nedi%CC%87r-10e24a1cc825

[22] https://en.wikipedia.org/wiki/Splash_screen

[23] https://en.wikipedia.org/wiki/Page_layout

[24] https://sirri-keles.medium.com/android-studio-2-2-19f832dc7467

[25] https://medium.com/@gizemcumen85/g%C3%B6r%C3%BCnt%C3%BC-i%CC%87%C5%9Fleme-teknolojisi-image-processing-262bb58fbb27

[26]ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-6.Hafta.pdf

[27] https://abdulsamet-ileri.medium.com/g%C3%B6r%C3%BCnt%C3%BC-filtrelerini-uygulama-ve-kenarlar%C4%B1-alg%C4%B1lama-21d42f194db4

[28]https://www.google.com/url?sa=i&url=https%3A%2F%2Fdergipark.org.tr%2Ftr%2Fdownload%2Farticle-file%2F326471&psig=AOvVaw1ovZonpgfNc4u23OJ6kU9b&ust=1718070345142000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCJDg-Pf0z4YDFQAAAAAdAAAAABAP

[29] https://e2eml.school/convert_rgb_to_grayscale

[30] https://bshopit.ibermaticoss.com/category?name=rgb%20to%20grayscale

[31]https://ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf

[32] https://tr.wikipedia.org/wiki/%C4%B0kili_g%C3%B6r%C3%BCnt%C3%BC

[33] N. Şahin and N. Alpaslan, "Cilt Lezyon Bölütlemesi için Metasezgisel Temelli Otsu Eşikleme Yöntemi", TDFD, vol. 9, no. 1, pp. 42–48, 2020, doi: 10.46810/tdfd.712911.

[34]https://tr.wikipedia.org/wiki/G%C3%B6r%C3%BCnt%C3%BC_e%C5%9Fikleme#cite_note-1

[35] https://en.wikipedia.org/wiki/Otsu%27s_method

[36] https://batuhandaz.medium.com/dijital-g%C3%B6r%C3%BCnt%C3%BC-i%CC%87%C5%9Fleme-image-processing-8-morfolojik-g%C3%B6r%C3%BCnt%C3%BC-i%CC%87%C5%9Fleme-b64bd50c012f

[37] https://medium.com/@sasasulakshi/opencv-morphological-dilation-and-erosion-fab65c29efb3

[38] https://pyimagesearch.com/2021/04/28/opencv-morphological-operations/

[39]https://en.wikipedia.org/wiki/Text_segmentation#:~:text=Text%20segmentation%20is%20the%20process,subject%20of%20natural%20language%20processing.

[40] https://towardsdatascience.com/segmentation-in-ocr-10de176cf373

[41] https://towardsdatascience.com/segmentation-in-ocr-10de176cf373

[42] https://hilalgozutok.medium.com/evri%C5%9Fimli-sinira%C4%9Flar%C4%B1-convolutional-neural-networks-cnn-e61470e9bdb1

[43] https://www.nomidl.com/natural-language-processing/resnet-explained/

[44] https://medium.com/@omerkaanvural/vanishing-gradient-problem-kaybolan-gradyan-sorunu-bba5634ef037#:~:text=K%C4%B1sa%20tan%C4%B1m%3A%20Bir%20derin%20

%C3%B6%C4%9Frenme,ya%C5%9Fanan%20%C3%B6%C4%9Frenme%20g%C3%BC%C3%A7l%C3%BC%C4%9F%C3%BCne%20VGP%20denir.

[45] https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd

[46] https://www.gtech.com.tr/yapay-sinir-aglari-ve-uygulamalari/#:~:text=Yapay%20Sinir%20A%C4%9Flar%C4%B1%20(YSA)%20dan%C4%B1%C5%9Fmanl%C4%B1,istenen%20%C3%A7%C4%B1k%C4%B1%C5%9F%20aras%C4%B1nda%20k%C4%B1yaslama%20yapmaktad%C4%B1r.

[47] https://www.saedsayad.com/artificial_neural_network_bkp.htm

[48] https://www.gtech.com.tr/yapay-sinir-aglari-ve-uygulamalari/#:~:text=Yapay%20Sinir%20A%C4%9Flar%C4%B1%20(YSA)%20dan%C4%B1%C5%9Fmanl%C4%B1,istenen%20%C3%A7%C4%B1k%C4%B1%C5%9F%20aras%C4%B1nda%20k%C4%B1yaslama%20yapmaktad%C4%B1r.

[49] https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

[50] https://miuul.com/blog/evrisimli-sinir-aglarina-giris

[51] https://hilalgozutok.medium.com/evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-convolutional-neural-networks-cnn-e61470e9bdb1

[52] https://tr.linkedin.com/pulse/deep-learningin-en-%C3%B6nemli-noktalar%C4%B1-ve-ipu%C3%A7lar%C4%B1-tunahan-%C3%B6zdo%C4%9Fan

[53] https://medium.com/@celikmehmetyl/streamlit-nedir-ve-ne-i%CC%87%C5%9Fe-yarar-a9ac57d3a3a4#:~:text=Streamlit%2C%20%C3%B6zellikle%20veri%20bilimi%20ve,h%C4%B1zla%20interaktif%20web%20uygulamalar%C4%B1na%20d%C3%B6n%C3%BC%C5%9Ft%C3%BCrebilirsiniz.

ÖZGEÇMİŞ

Şule Meşe Cumhuriyet Anadolu Lisesi'nde lise eğitimini tamamladıktan sonra Bandırma Onyedi Eylül Üniversitesi Bilgisayar Mühendisliği bölümünde eğitimine devam etmektedir. Üniversite eğitimi sırasında Beyçelik Gestamp A.Ş ve TOFAŞ A.Ş. gibi firmalarda staj yapma fırsatı bulmuştur.