KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE EL YAZISI TANIMA

YÜKSEK LİSANS TEZİ

Alperen UZUN

EKİM 2024
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE EL YAZISI TANIMA

Alperen UZUN
ORCID : - - -

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

Tezin Enstitüye Verildiği Tarih : 19 / 09 /2024
Tezin Savunma Tarihi : 28 / 10 /2024

Tez Danışmanı : Dr. Öğr. Üyesi Mehmet ÖZTÜRK
ORCID : - - -

Trabzon 2024

ÖNSÖZ

"Derin öğrenme yöntemleri ile Türkçe el yazısı tanıma" isimli bu tez Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektronik Mühendisliği Yüksek Lisans Programı'nda hazırlanmıştır.

Tez çalışma sürecimde hiçbir desteğini esirgemeyen, bilgi ve tecrübelerinden yararlandığım değerli tez danışman hocam Dr. Öğr. Üyesi Mehmet ÖZTÜRK'e teşekkürü bir borç bilirim. Yüksek lisans eğitimim boyunca deneyimleri ve öğretileriyle akademik ve kişisel gelişimime katkıda bulunan değerli hocalarıma teşekkür ederim.

Çalışmalarım süresince desteklerini benden esirgemeyen Elektrik-Elektronik Mühendisliği Bölümü'ndeki tüm değerli hocalarıma ve asistan arkadaşlarıma içten teşekkürlerimi sunarım.

Bütün hayatım boyunca beni her konuda destekleyen, her zaman yanımda olan ve bugünlere gelmemde en büyük paya sahip annem Şenay UZUN, babam Yaşar UZUN ve kardeşim Pınar Çağla UZUN'a teşekkür ederim.

Alperen Uzun
Trabzon 2024

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Derin öğrenme yöntemleri ile Türkçe el yazısı tanıma” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Mehmet ÖZTÜRK’ün sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 28/10/2024

Alperen UZUN

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE EL YAZISI TANIMA

Alperen UZUN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Mehmet ÖZTÜRK
2024, 51 Sayfa

El Yazısı Karakter Tanıma, karakterleri bir görüntü olarak algılayıp bu görüntüleri analiz ederek metin formatında işleyebilen algoritmalar kullanır. Bu teknoloji belgelerin dijitalleştirilmesi ve arşivlenmesi gibi birçok alanda önemli bir rol oynar. Literatürde el yazısı karakter tanıma çalışmaları genellikle İngilizce alfabe ve yaygın olarak kullanılan Latin karakterleri üzerine yoğunlaşmaktadır. Türkçe gibi özel karakterler içeren diller için geliştirilmiş veri setlerinin sınırlı olması bu dillerdeki tanıma sistemlerinin performansını olumsuz yönde etkileyebilir. Bu çalışmanın hedefi Türk alfabesindeki özel karakterleri içeren veri setleri hazırlayarak el yazısı karakter tanıma alanında literatüre katkı sağlamak ve kendi oluşturduğumuz bir veri seti üzerinde derin öğrenme tekniklerinin başarımını değerlendirmektir. Bu amaçla bir kurumda çalışan 400 kişinin farklı yazım stilleriyle gönüllü olarak doldurduğu belgelerden el yazısı karakterler elde edilerek bir veri seti oluşturulmuş, ardından derin öğrenme tabanlı CNN (Evrişimsel Sinir Ağı) modelleri bu veri setiyle eğitilmiştir. CNN modellerinin test sonuçları karşılaştırıldığında en başarılı modelin %93,86 test başarısına ulaştığı görülmüştür. En başarılı CNN modelininin performansı çeşitli metriklerle değerlendirilmiş ve modelin sınıflandırma başarısı detaylı bir şekilde analiz edilmiştir.

Anahtar Kelimeler: El Yazısı Karakter Tanıma, Derin Öğrenme, Evrişimsel Sinir Ağları (CNN), Türk Alfabesi

MSc. Thesis

SUMMARY

TURKISH HANDWRITING RECOGNITION WITH DEEP LEARNING METHODS

Alperen UZUN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Graduate Program
Supervisor: Assist. Prof. Dr. Mehmet ÖZTÜRK
2024, 51 Pages

Handwritten Character Recognition uses algorithms that can detect characters as images and analyze these images to process them in text format. This technology plays a significant role in various fields, such as digitizing and archiving documents. In the literature, studies on handwritten character recognition generally focus on the English alphabet and commonly used Latin characters. The limited availability of datasets developed for languages with special characters, such as Turkish, can negatively affect the performance of recognition systems in these languages. The aim of this study is to contribute to the literature in the field of handwritten character recognition by preparing datasets containing special characters from the Turkish alphabet and to evaluate the performance of deep learning techniques on our own dataset. For this purpose, a dataset was created by extracting handwritten characters from documents voluntarily filled out by 400 employees in an institution, each with different writing styles. Then, deep learning-based CNN (Convolutional Neural Network) models were trained with this dataset. When the test results of the CNN models were compared, it was found that the most successful model achieved a test accuracy of 93.86%. The performance of the best CNN model was evaluated using various metrics, and the classification success of the model was analyzed in detail.

Key Words: Handwriting Character Recognition, Deep Learning, Convolutional Neural Networks (CNN), Turkish Alphabet

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER VE KISALTMALAR DİZİNİ

CNN : Evrişimli Sinir Ağı
HCR : El Yazısı Karakter Tanıma
SVM : Destek Vektör Makinesi
ANN : Yapay Sinir Ağı
DCNN : Derin Konvolüsyonel Sinir Ağı
MNIST : Değiştirilmiş Ulusal Standartlar ve Teknoloji Enstitüsü
ResNet : Artık Bağlantılı Ağ
CAR : Karakter Doğruluk Oranı
OCR : Optik Karakter Tanıma
RNN : Tekrarlayan Sinir Ağları
K : Filtre boyutu
S : Kaydırma
Epok : Eğitim süreci
RGB : Kırmızı Yeşil Mavi
R : Kırmızı bişelen değeri
G : Yeşil bileşen değeri
B : Mavi bileşen değeri
σ : Sigma
G_x : Yatay gradyan
G_y : Düşey gradyan
θ : Theta
ρ : Rho
m : Doğrusal eğim
c : Doğrunun y eksenini kestiği nokta
L : Hough Doğru Dönümü çizgi uzunluğu

1. GİRİŞ

El yazısı her bireye özgü bir özelliktir ve tıpkı DNA veya parmak izi gibi kişisel bir kimlik unsuru taşır. Bu benzersizlik her bireyin yazı stili, kalemi tutuş şekli ve kağıda uyguladığı basınç gibi değişkenlerle belirlenir. El yazısının bu bireysel farklılıkları modern teknolojilerle birleştiğinde özellikle imza doğrulama, adli inceleme ve otomatik karakter tanıma alanlarında daha da değer kazanmaktadır (Oak vd., 2021).

El Yazısı Karakter Tanıma (HCR) dijital dünyada el yazısıyla yazılmış metinlerin anlaşılması ve otomatik olarak okunmasını sağlayan önemli bir teknolojidir. Bu teknoloji özellikle belgelerin dijitalleştirilmesi, arşivlenmesi, veri girişi otomasyonu ve belge işleme süreçlerinde yaygın olarak kullanılmaktadır. Son yıllarda el yazısı karakter tanıma alanında derin öğrenme yöntemleri büyük ilerlemeler kaydetmiştir. Bu alandaki en etkili yaklaşımlardan biri evrişimli sinir ağlarıdır (CNN). CNN modelleri karmaşık ve çeşitli el yazısı karakterlerini yüksek doğrulukla tanıma yeteneğine sahiptir. Ancak bu modellerin başarılı olabilmesi için geniş ve çeşitli veri kümelerine ihtiyaç duyulmaktadır.

Literatürde yapılan çalışmalar incelendiğinde el yazısı karakter tanıma çalışmalarının genellikle İngilizce ve Latin alfabesi üzerine yoğunlaştığı görülmektedir. Türk alfabesine özgü harflerin ("ğ", "Ğ", "ç", "Ç", "ş", "Ş", "ü", "Ü", "ö", "Ö", "ı", "İ") yer aldığı veri setlerinin sınırlı olması Türkçe el yazısı karakterlerinin tanınması konusunda zorluk oluşturmaktadır. Bu çalışmanın amacı Türk alfabesindeki karakterleri içeren bir veri seti oluşturarak mevcut literatürdeki bu boşluğu doldurmak ve el yazısı karakter tanıma sistemlerinin performansını iyileştirmektir. Derin öğrenme tekniklerinin bu sürece entegrasyonu tanıma sistemlerinin doğruluğunu ve verimliliğini önemli derecede artırmaktadır. Bu çalışmada derin öğrenme tabanlı CNN modellerini eğitmek amacıyla Türk alfabesine özgü harfleri içeren bir veri seti hazırlanmıştır. Farklı katmanlar ve parametreler içeren CNN modelleri bu veri seti ile eğitilerek modeller test edilmiş ve en yüksek sınıflandırma başarısına sahip olan CNN modeli elde edilmeye çalışılmıştır. Ayrıca veri setlerine çeşitli ön işleme ve veri artırma teknikleri uygulanarak modellerin doğruluğu artırılmaya çalışılmıştır.

2. LİTERATÜR ARAŞTIRMASI

2.1. El Yazısı Karakter Tanıma (HCR) Sistemlerine Bakış

HCR, dijital ortama aktarılan el yazısı karakterlerin otomatik olarak tanınması ve metin haline getirilmesi sürecidir. Bu teknoloji karakterleri bir görüntü olarak algılayan ve bu görüntüleri analiz ederek metin formatında işleyebilen algoritmalar kullanır (Shah ve Jethava, 2013). HCR teknolojisi bilgisayarla görme alanında ele alınan en eski problemlerden biridir ve bu konuda yapılan araştırmalar uzun yıllardır devam etmektedir (Kimura vd., 1987; Kang vd., 2021). Bu alandaki ilk önemli çalışmalardan biri Grimsdale ve diğerleri (1959) tarafından gerçekleştirilmiştir. Bu dönemde bilgisayar bilimlerinin gelişimiyle birlikte karakter tanıma üzerine birçok çalışma yapılmaya başlanmıştır. Eden'in (1968) önerdiği sentezle analiz yöntemi karakter tanıma araştırmalarında bir dönüm noktası olmuştur. Eden'in çalışması el yazısı karakterlerin sınırlı sayıda şematik özellikten oluştuğunu resmi olarak kanıtlamıştır. Bu fikir daha sonraki yapısal karakter tanıma yaklaşımlarında temel bir rol oynamış ve karakterlerin çeşitli dil ve yazı sistemlerinde nasıl tanınabileceğine dair birçok yöntemin geliştirilmesine yol açmıştır. El yazısı karakter tanıma üzerine yapılan çalışmalar daha çok basit karakter setleri ve sabit yazı stilleri üzerine yoğunlaşmışken 1990'lı yıllardan itibaren daha karmaşık el yazısı tarzlarını tanımaya yönelik yöntemler geliştirilmiştir (Suen ve Nadal, 1990; Plamondon ve Srihari, 2000). Yapay sinir ağları ve daha sonra derin öğrenme tekniklerinin devreye girmesiyle karakter tanıma sistemleri giderek daha doğru ve verimli hale gelmiştir (Graves ve Schmidhuber, 2009; Liu vd., 2004).

Günümüzde HCR sistemleri yalnızca basit karakter tanımanın ötesine geçerek karmaşık dil ve yazı sistemlerini destekleyebilmektedir. Özellikle dil özelliklerine özgü modeller geliştirilmiş ve farklı dillerdeki karakter setlerinin tanınmasını kolaylaştırmıştır (Vidhale vd., 2021). Fakat HCR teknolojisinin hala karşılaştığı zorluklar bulunmaktadır. El yazısındaki kişisel varyasyonlar, dilsel farklılıklar ve metinlerin karmaşık yapıları model performansını etkileyen önemli faktörlerdir. Bu zorlukların üstesinden gelmek için araştırmacılar sürekli olarak yeni yöntemler ve teknikler geliştirmekte ve bu bağlamda veri çeşitliliğini artırmak, model genelini iyileştirmek için çalışmalar yapmaktadır (AlKendi vd., 2024).

2.2. İlgili Çalışmalar

Derin öğrenme gelişmeleri el yazısı karakter tanıma sistemlerinin evriminde önemli bir rol oynamıştır. Bu teknolojik gelişmeler özellikle el yazısı belgelerin tanınmasında yaygın olarak kullanılan yapay sinir ağlarının başarısını artırmıştır. Bu bağlamda en yaygın ve en gelişmiş yöntemlerden biri olan Evrişimli Sinir Ağları (CNN), el yazısı karakter tanıma görevlerinde öne çıkmaktadır. CNN tabanlı yaklaşımlar potansiyel olarak sınırlı bir sürede konumla değişmeyen özellikler hesaplayabilme yeteneklerine sahiptir. Bu özelliği sayesinde CNN'lerin bozulmalara karşı diğer el yazısı metin tanıma yöntemlerine kıyasla daha avantajlı olduğu söylenebilir (Agrawal vd., 2024).

Bu avantajların ne denli önemli olduğunu anlamak için HCR alanında yapılan çalışmalar incelenmelidir. Son yıllarda CNN'ler ve diğer derin öğrenme teknikleri üzerine yapılan çalışmalar bu ağların MNIST veri setindeki performanslarını kapsamlı bir şekilde ele almıştır. MNIST veri seti LeCun ve diğerleri tarafından (1998) tanıtılmış olup aynı çalışmada CNN'lerin de sunulmuş olması bu yöntemlerin el yazısı karakter tanımada ne kadar etkili olduğunu göstermektedir. Yapılan çalışmalar genellikle en yüksek başarı oranlarının CNN'ler kullanılarak elde edildiğini ortaya koymuştur.

Bhatt ve Patel (2018) karakter tanıma için kullanılan Destek Vektör Makinesi (SVM), Yapay Sinir Ağı (ANN), Konvolüsyonel Sinir Ağı (CNN), Derin Konvolüsyonel Sinir Ağı (DCNN) gibi çeşitli sınıflandırıcı yöntemlerini karşılaştırmış ve analiz etmiştir. Çalışmaları derin öğrenme yöntemlerinin karakter tanıma konusunda yüksek doğruluk sağladığını ortaya koymuştur. Bir başka çalışmada Das ve arkadaşları (2018) BANGLA el yazısı karakter tanıma için kullanılan popüler CNN mimarilerini çekirdek boyutu, havuzlama yöntemleri ve aktivasyon fonksiyonları ile birlikte incelemiştir. Çalışmanın sonuçları CNN mimarisinin karakter tanıma doğruluğunu artırabileceğini kanıtlamıştır.

Derin öğrenme teknikleri el yazısı metin tanıma amacıyla görüntülerin belirli özelliklerine odaklanarak tanıma işleminin daha etkin ve doğru olmasını sağlayabilir. Rajyagor ve Rakhlia'nın çalışmalarında, farklı dillerden taranan el yazısı karakterlerin tanınması için kullanılan derin öğrenme yöntemleri karşılaştırılmıştır. Bu çalışmada karakter tanıma için kullanılan yaklaşımların başarı oranlarının %75 ile %98 arasında değiştiği gözlemlenmiştir. Sonuç olarak daha yüksek doğruluk oranlarına ulaşmak ve karakter tanıma hızını artırmak için CNN'lerin derin öğrenme yöntemleriyle birlikte kullanılması gerektiği vurgulanmıştır (Rajyagor ve Rakholia, 2020).

El yazısı karakter tanıma sistemlerinin başarısını etkileyen en önemli faktörlerden biri modelin öğrenme kapasitesini artıracak çeşitlilik ve yeterli sayıda örnek içeren büyük ve dengeli bir veri setine sahip olmaktır. Çin dili için el yazısı karakterlerini içeren CASIA veri setiyle yapılan bir çalışmada yaklaşık 1,4 milyon etiketli karakter kullanılmıştır (Liu vd., 2011). Bir başka çalışmada kullanılan veri seti MADCAT, toplamda 750.000 segmentlenmiş Arapça belgelerden oluşur (Shtaiwi vd., 2022).

Bu bağlamda veri artırma teknikleri modelin performansını geliştirmek için stratejik bir şekilde uygulanmalıdır. Örneğin Luo ve diğerleri optik model eğitimi için geliştirdikleri uyarlanabilir veri artırma yöntemi ile modelin eğitim sürecine göre görüntüleri daha zor hale getirerek öğrenme kapasitesini artırmayı hedeflemiştir (Luo vd., 2020). Benzer şekilde Eltay ve diğerleri veri setindeki karakterlerin frekans dağılımına dayanan bir veri artırma yöntemi önermiştir. Bu yöntemle sınıflar arasında dengesizlik olduğunda daha az görülen karakterlere daha fazla ağırlık verilerek dengesizlikler giderilmeye çalışılmıştır (Eltay vd., 2021).

Bir başka çalışmada Gao ve diğerleri dijital el yazısı fontlarının tanınmasında CNN tabanlı modellerinin doğruluğunu artırmak amacıyla Gauss gürültüsü ve renk bozulması gibi veri artırma teknikleri uygulamıştır. Ayrıca modelde gereksiz parametrelerin eğitim üzerindeki etkisini azaltmak için ayrılabilir evrişim kullanılmıştır. Modelin eğitimi sırasında bozulmayı önlemek için ise ResNet'in kısa bağlantıları devreye sokulmuştur. MNIST veri kümesi üzerinde yapılan bu çalışmada, %99,82 gibi oldukça yüksek bir doğruluk oranına ulaşılmış ve modelin test performansında önemli bir iyileşme kaydedilmiştir (Gao vd., 2022).

El yazısı karakterlerde kullanılan veri artırma yöntemleri genellikle görüntü bazlıdır ve tüm görüntü üzerinde değişiklikler yapar. Hayashi ve arkadaşları döndürme, gürültü ekleme ve erozyon gibi görüntü işleme yöntemleri uygulayarak her el yazısı karakter için 50 karakter görüntüsünden 450 karakter görüntüsü üretmiştir (Hayashi vd., 2018).

Narayanan el yazısı karakterlerde kullanılan geleneksel görüntü bazlı veri artırma yöntemlerinin aksine veri odaklı bir yöntem önermiştir. Bu yöntem karakterlerin görünümünü değiştirmeye odaklanır. Genellikle karakterlerin şekilleri, kalınlıkları veya eğrilikleri değiştirilir. El yazısına özgü değişiklikler yapılarak karakterlerin çeşitli varyasyonları arttılır. Bu sayede el yazısı veri setlerinde daha spesifik ve etkili bir veri artırmı sağlanmıştır (Narayanan, 2023).

Hidayat ve diğerleri HSPLM veri setindeki sınırlı veri örnekleri sorununu çözmek için veri artırma teknikleri kullanmıştır. Geometrik dönüşümler yanında aynı zamanda görüntü arka planına gürültü ekleme ve parlaklık ayarlama gibi yöntemler de kullanarak veri çeşitliliğini artırmışlardır. Bu yöntemler uygulanırken veri setindeki karakterlerin dengesini sağlamaya dikkat edilmiştir. Bu işlemlerin sonucunda CAR (Karakter Doğruluk Oranı) %97.4 ile kayda değer bir performans göstermiştir (Hidayat vd., 2021).

HCR sistemlerinde veri artırma tekniklerinin yanı sıra kullanılan ön işleme yöntemleri de verinin doğruluğunu ve modelin performansını artırmak amacıyla önemlidir. Veri ön işleme ham veriyi analiz ve modelleme için daha uygun bir formata dönüştürme sürecidir. Bu adım veri biliminde önemli bir aşamadır çünkü verinin kalitesi modelin başarısını doğrudan etkiler. Literatürde birçok çalışmada farklı ön işleme teknikleri kullanılmıştır. Trier ve diğerleri (1996) el yazısı karakter tanıma sistemlerinde Gaussian filtresi kullanarak görüntüdeki gürültüyü azaltma tekniği uygulamıştır. Bu işlem özellikle zayıf kontrastlı veya bozulmuş görüntülerde karakter sınırlarının belirginleştirilmesine yardımcı olmuştur. Bir başka çalışmada Caesar ve diğerleri (1993) veri setine normalizasyon teknikleri uygulayarak görüntülerdeki karakterlerin ve kelimelerin çeşitli değişkenliklerini standart hale getirmişlerdir. Bu sayede tanıma sistemi daha tutarlı ve doğru sonuçlar vermiştir.

Liu ve Marukawa el yazısı karakter tanıma sürecinde normalizasyon ve gürültü azaltma tekniklerine odaklanmışlardır. Çalışmada karakterlerin boyut ve konum varyasyonlarını minimize etmek için ön işleme adımları olarak gradyan tabanlı normalizasyon ve adaptif gürültü azaltma yöntemleri kullanılmıştır. Bu teknikler karakterlerin daha tutarlı bir şekilde tanınmasını sağlayarak modelin doğruluğunu artırmıştır (Cheng-Lin Liu ve Marukawa, 2004)

Diğer bir ön işleme yöntemi olan Otsu algoritması, el yazısı karakter tanıma işlemlerinde görüntü binarizasyonunu otomatikleştirmek ve metin ile arka plan arasındaki ayrımı optimize etmek için etkili bir araçtır. Paul ve Tunga, Otsu algoritmasını temel alan bir binarizasyon tekniği önermiştir. Doküman görüntülerinin kötü olduğu durumlarda toleranslı hale getirmek için Otsu algoritmasının üzerinde iyileştirme yapılmıştır. Bu yöntem ile zorlu koşullarda bile ön plan ile arka planı etkili bir şekilde ayırabilmek hedeflenmiştir. H-DIBCO2012 ve DIBCO2009 veri setlerinde test edilen bu yöntem deneysel sonuçlara göre Otsu algoritmasından daha yüksek bir hassasiyet sağlamış ve daha iyi sonuçlar vermiştir (Paul ve Tunga, 2016).

El yazısı karakter tanıma çalışmalarında ayrıca dil ve görev çeşitliliği de artmaktadır. Kılıçlar ve Makinacı, C programlama dilinde yazılmış kaynak kodlarını içeren sınav kağıtları üzerinde el yazısı karakter tanıma gerçekleştirmiştir. Bu çalışmada taranmış sınav kağıtlarına önişleme adımları uygulanmış ve belge görüntülerine filtreleme ile yönlendirme düzeltmesi yapılmıştır. Ardından bölütleme ve normalizasyon teknikleriyle veri temsilleri elde edilmiş ve CNN'lerin eğitiminde kullanılmıştır. Sonuç olarak el yazısı kaynak kodu karakterlerinin tanınması için CNN tabanlı sınıflandırıcıların %92,33 ile %98,82 arasında başarılı sonuçlar verdiği gözlemlenmiştir (Kılıçlar ve Makinacı, 2019).

Genel olarak derin öğrenme tekniklerinin HCR alanındaki gelişimi el yazısı karakter tanıma sistemlerinin daha doğru ve hızlı çalışmasına katkıda bulunmuştur. CNN gibi modern mimariler, veri artırma ve ön işlenme teknikleri ile el yazısı tanıma görevlerinde büyük başarılar elde edilmesine olanak tanımaktadır. Bu alanda yapılan çalışmalar derin öğrenmenin gelecekte de el yazısı tanıma sistemlerinin gelişimine yön vereceğini göstermektedir.

3. TEORİK ARKAPLAN

3.1. El Yazısı Karakter Tanıma

Günlük yaşamda el yazısı belgeler iletişim kurma ve bilgileri saklama amacıyla yaygın olarak kullanılmaktadır. HCR el yazısı ile yazılmış belgelerin dijital olarak taranmış görüntülerinden harflerin otomatik olarak tanınması sürecidir. HCR sistemlerinde başlıca problem farklı yazarlar için tamamen farklı olabilen el yazısı stillerinin çeşitliliğidir (Patel ve Thakkar, 2015). HCR teknolojisi OCR (Optik Karakter Tanıma) ile benzerlik gösterse de HCR el yazısının karmaşıklığı ve çeşitliliği nedeniyle daha gelişmiş algoritmalar ve modeller kullanır (Nikitha vd., 2020).

El yazısı karakter tanıma sistemleri çevrimiçi karakter tanıma ve çevrimdışı karakter tanıma olmak üzere iki kategoriye ayrılır. Çevrimiçi karakter tanıma sistemlerinde karakterler oluşturulma sürecinde tanıma yapılır. Çevrimdışı karakter tanımada ise ilk olarak el yazısı belgeler oluşturulur, taranır, bilgisayara kaydedilir ve ardından tanıma yapılır (Vinjit vd., 2020). Çevrimdışı el yazısı karakter tanıma birçok araştırma tarafından ele alınan en eski bilgisayarla görme problemlerinden biri olarak kabul edilmektedir (Peng vd., 2019). Bu alandaki sürekli ilgi el yazısı tanımanın karmaşıklığı, yazı stillerinin çeşitliliği, büyük karakter veri setleri ve metinlerin karmaşık yapıları gibi faktörlerden kaynaklanır. El yazısı tanıma teknolojisi sadece genel el yazısı tanıma ile sınırlı kalmayıp aynı zamanda banka formlarının otomatik olarak tanınması ve tarihi belgelerin işlenmesi gibi özel alanlarda da kullanılmaktadır. Şekil 1'de çevrimdışı el yazısı karakter tanıma akış şeması gösterilmiştir. Tipik bir HCR sistemi üç ana bölümden oluşur:

- Ön işleme
- Özellik çıkarma
- Sınıflandırma

Şekil 1. Çevrimdışı el yazısı karakter tanıma akış şeması

3.1.1. Ön İşleme

Ön işleme aşaması HCR sistemlerinde kritik bir rol oynar. Ham görüntü verileri modelin doğruluğunu ve genel performansını etkileyebilecek birçok bozulma ve gürültü içerebilir. Görüntülerde ön işleme aşaması ham görüntüler üzerinde çeşitli işlemler yaparak bu görüntülerin analiz ve modelleme için daha uygun hale getirilmesini sağlayan bir süreçtir. Bu süreç görüntüdeki önemli bilgileri öne çıkarabilir, görüntülerin kalitesini ve modelin doğruluğunu artırabilir (Chaki ve Dey, 2018).

HCR sistemlerde verilerin ön işlenmesindeki amaç veri toplama sürecindeki eksiklikleri düzeltmek ya da veriyi sonraki aşamalarda kullanılacak şekilde hazırlamaktır. Bu sayede sınıflandırmayı karmaşıklaştıracak ve tanıma oranını azaltacak varyasyonlar azaltılır (Neetu Rani, 2017). Bu aşama karakterlerin daha belirgin ve tutarlı bir şekilde temsil edilmesini sağlamak amacıyla verilerin iyileştirilmesi için bir dizi teknik içerir. HCR sistemlerde genel olarak kullanılan veri ön işleme teknikleri aşağıda verilmiştir:

- Görüntü Normalizasyonu
- Gürültü Giderme
- İkili Görüntüleme

Görüntü Normalizasyonu: Görüntü işleme bağlamında normalizasyon, görüntülerin boyutlarını, renk değerlerini veya diğer özelliklerini belirli bir formata veya aralığa getirmek anlamına gelir. Bir sinir ağı modelinin belirli bir boyutta giriş alması gerekiyorsa ve veri kümesinde farklı boyutta görüntüler bulunuyorsa bu görüntülerin boyutlarını değiştirmeden ya da bozmadan eşit bir boyuta getirmek gerekir. Bu özellikle CNN gibi modellerde önemlidir çünkü modelin giriş boyutunun sabit olması gerekir. Bu sayede modelin tutarlı boyutta veri alması sağlanır ve öğrenme süreci iyileştirebilir (Alginahi, 2010) Balci ve diğerleri el yazısı kelimeleri sınıflandırma amacıyla yaptıkları çalışmada genişlik ve yükseklik farklılıkları nedeniyle görüntüleri aynı boyuta getirmek için kırpma veya yeniden ölçekleme yöntemlerini önermemiştir. Çünkü kırpma bazı karakterlerin kaybolmasına ve yanlış yorumlamalara yol açabilirken, yeniden ölçekleme görüntülerin bozulmasına neden olabilir. Bu nedenle tüm görüntüleri veri setindeki maksimum genişlik ve yükseklikte beyaz boşluk ekleyerek padding (doldurma) yapmaya karar verilmişlerdir. Bu yöntem ile birlikte görüntülerin temel özellikleri korunmuş ve görüntüler arasındaki ilişkiler değişmeden boyut uyumu sağlanmıştır (Balci vd., 2017).

Gürültü Giderme: Görüntülerde gürültü, piksel değerlerindeki hatalar veya çıktıda herhangi bir önemi olmayan istenmeyen desenler olarak tanımlanır. Gürültü görüntünün edinim süreci sırasında ortaya çıkabilir ve çizgilerde boşluklar, kesik segmentler ve belgede dolu döngüler oluşturabilir (Bansal ve Kumar, 2013). Düşük kaliteli belgelerde genellikle tuz ve karabiber gürültüsü ortaya çıkar ve izole pikseller, açık bölgelerdeki pikseller ya da kapalı bölgelerdeki kapalı pikseller şeklinde görünür (Kannan vd., 2008). Bu tür gürültü filtreleme yöntemleri kullanılarak temizlenebilir. Filtreleme bir pikselin değerini çevresindeki piksellerin değerlerine dayalı olarak belirleyen bir yöntemdir. Bu işlem düzensiz yazı yüzeyleri veya düşük kaliteli veri toplama cihazlarından kaynaklanan istenmeyen desenleri azaltır ve hafif dokulu arka planları temizleyip görüntüyü netleştirir. Gürültüyü gidermek için tasarlanan filtrelere Gauss filtreleme, medyan filtreleme, iki yönlü filtreleme gibi filtreleme teknikleri örnek verilebilir (Tin, 2011).

İkili Görüntüleme: HCR sistemlerinde ikili görüntüleme yöntemi karakterlerin tanınabilirliğini artırmak için sıklıkla kullanılan bir ön işleme tekniğidir. Gri tonlamalı veya renkli bir görüntüyü iki renkten oluşan bir ikili görüntüye dönüştürür. Bu yöntem görüntüdeki ön plan ve arka plan arasında net bir ayrım yaparak karakterlerin ve diğer önemli öğelerin daha belirgin hale gelmesini sağlar (Sauvola vd., 1997). Bu işlem genellikle eşikleme teknikleri kullanılarak gerçekleştirilir. Pikseller eşik değerinin üstünde olup olmamasına göre siyah (0) veya beyaz (255) olarak atanır.

3.1.2. Özellik Çıkarma

Özellik çıkarımı aşamasında metin görüntülerinden kritik özellikler elde edilir ve bu özelliklerin sınıflandırma sisteminin performansını iyileştirmek için kullanılabilir olması sağlanır. Bu aşama verinin analizi ve modelleme aşamaları için temel bir adımdır (Jogin vd., 2018). Özelliklerin sınıflandırma görevleriyle ilişkili olup olmadığına göre iki ana yöntem türü vardır. El yapımı yöntemler özelliklerin konumunu ve önemini önceden belirlemeyi gerektirir. Bu nedenle sınırlı bir uygulama alanına sahiptir ve el yazısı stillerindeki değişiklikler, arka plan renkleri ve diğer düzensizlikler nedeniyle hassas olabilir. Otomatik öğrenme yöntemleri ise derin sinir ağı mimarilerini kullanarak özellikleri otomatik olarak öğrenir. Bu yöntemler çeviri, ölçekleme ve bozulma gibi sorunları çözerek daha dayanıklı sistemler oluşturur.

3.1.3. Sınıflandırma

Sınıflandırma aşamasında temsilci özellikler eğitilmiş bir sınıflandırıcıya aktarılır ve sınıflandırıcı karakter veya kelime sınıfını tahmin eder. Metin görüntüsü tanıma yaklaşımları genellikle iki ana kategoriye ayrılır:

Segmentasyonlu yaklaşım: Geleneksel bir yöntem olup metin satırı görüntüsünü bireysel karakterlere ayırarak ardından eğitilmiş bir sınıflandırıcı kullanarak her karakteri tanımayı içerir. Bu yöntemin etkinliği karakter segmentasyonunun doğruluğuna büyük ölçüde bağlıdır ve herhangi bir segmentasyon hatası sınıflandırıcının tanıma doğruluğunu doğrudan etkiler.

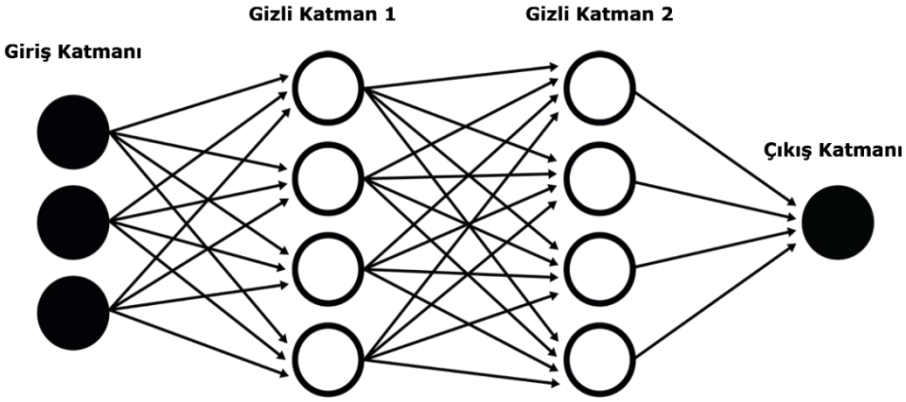
Segmentasyonsuz yaklaşım: Kelime, satır veya çoklu satır görüntülerini doğrudan işleyerek açık bir segmentasyon yapmadan tanıma gerçekleştirir. Arka plan karmaşıklığı, bitişik metin satırları veya üst üste binen karakterler gibi ayrımların belirlenmesinin zor olduğu durumlarda özellikle yararlıdır. Bu tür yöntemlerde sıklıkla kullanılan teknikler arasında Konvolüsyonel Sinir Ağları (CNN) ve Tekrarlayan Sinir Ağları (RNN) bulunur. Bu yöntemler tanıma performansını artırmak için geçmiş ve gelecekteki karakterlerden veya kelimelerden elde edilen bağlamsal bilgiyi kullanır (Retsinas vd., 2023).

3.2. Derin Öğrenme

Derin öğrenme makine öğreniminin bir alt kümesidir ve günümüzde makine öğrenimi alanında en yaygın kullanılan ve başarılı sonuçlar elde eden tekniklerden biridir. Özellikle görüntü tanıma, desen tanıma ve doğal dil işleme gibi çeşitli uygulama alanlarında büyük başarılar sağlamıştır (Hosseini vd., 2019).

Derin öğrenme yapay sinir ağları kullanarak çalışır. Bu ağlar beynimizdeki sinir hücrelerinin (nöronlar) birbirleriyle bağlantı kurma ve bilgi işlem şeklini model alır (Sharifani ve Amini, 2023). İnsan benzeri sonuçlar çıkarmayı hedefleyerek verileri sürekli olarak belirli bir mantıksal yapıyla analiz eder.

Bu sinir ağları çok katmanlı bir algoritma yapısından oluşur. Önceki katmanın girişini bir sonraki katmanın çıkışı olarak kullanarak yüksek derecede soyutlanmış veri özellikleri öğrenilir (Hao, 2019). Sinir ağlarının katmanları kaba verilerden ayrıntılı verilere doğru çalışan bir tür filtre olarak düşünülebilir. Şekil 2’de Derin Öğrenme Sinir Ağı Mimarisi gösterilmiştir.



Şekil 2. Derin Öğrenme Sinir Ağı Mimarisi

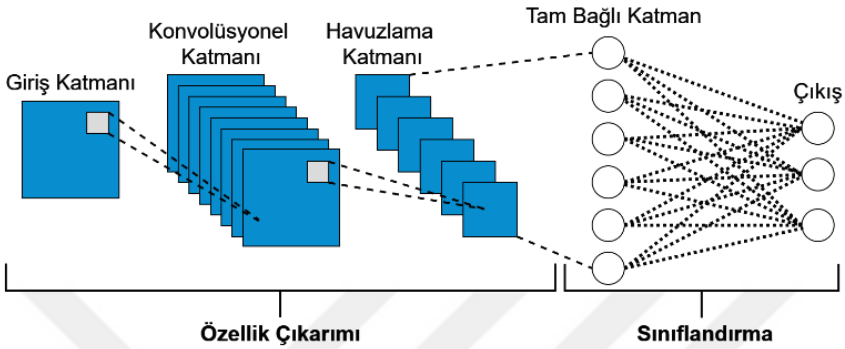
Tipik bir sinir ağı mimarisi birkaç katmandan oluşur:

1. Giriş Katmanı: İlk katman giriş katmanıdır. Giriş katmanı genellikle verinin ham hali veya ön işlenmiş veriyi temsil eder. Her bir düğüm (nöron) genellikle bir özellik veya veri noktası ile ilişkilidir (Du vd., 2016).
2. Gizli Katmanlar: İlk gizli katman giriş verilerini daha soyut bir düzeyde işlerken ikinci gizli katman bu soyut özellikleri daha da karmaşık hale getirir ve daha yüksek seviyeli özellikler öğrenir. Gizli katmanların sayısı ve her bir katmandaki nöron sayısı ağın kapasitesini ve karmaşıklığını belirler (LeCun vd., 2015).
3. Çıkış Katmanı: Sinir ağının sonucunu verdiği son katmandır. Gizli katmanlarda öğrenilen özellikler kullanılarak belirli bir görev için sonuçlar üretilir. Örneğin sınıflandırma görevlerinde çıkış katmanı her bir sınıfa ait olasılıkları temsil eder. Regresyon görevlerinde ise sürekli bir değer sağlar (Sewak vd., 2020).

3.3. Evrişimli Sinir Ağları (CNN)

Evrişimli sinir ağı (CNN), derin öğrenme alanındaki en önemli ağlardan biridir. Bilgisayarla görü ve doğal dil işleme gibi birçok alanda etkileyici başarılar elde etmesinden dolayı son birkaç yılda hem endüstri hem de akademide alanında büyük ilgi görmüştür (Li vd., 2022). Geleneksel yapay sinir ağları ile arasındaki en belirgin fark özellikle görüntülerdeki desenleri tanıma konusunda daha iyi performans göstermeleridir. Bu özelliğinden dolayı el yazısı karakter tanıma görevlerinde karakterlerin farklı yazım stillerine, boyutlarına ve açısal farklılıklarına rağmen ortak özellikleri öğrenme ve tanıma yeteneği sayesinde geleneksel yöntemlere görey daha yüksek doğruluk oranları sağlar.

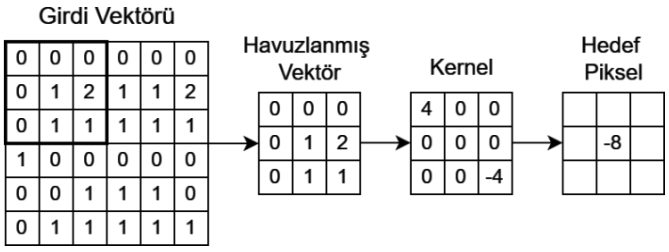
CNN modelleri genellikle konvolüsyonel katmanı, havuzlama katmanı ve tam bağlı olmak üzere üç ana katmandan oluşur. Bu katmanlar giriş verisinin dönüştürüldüğü ve nöron birimlerinin organize olduğu düzlemlerdir. Bu katmanlar üst üste konulduğunda Şekil 3'te gösterilen bir CNN mimarisi oluşur.



Şekil 3. CNN mimarisi

3.3.1. Konvolüsyonel Katman

Konvolüsyonel katman CNN mimarisinin temel bileşenidir. Bu katman parametreleri genellikle kare olan ve her birinin belirli bir genişlik ve yüksekliği bulunan K öğrenilebilir filtre kümesini içerir. Bu filtreler hacmin tam derinliği boyunca uzanır. Şekil 4'te görsel temsili verilen konvolüsyonel katmanda çekirdeğin merkez bileşeni giriş verisinin (örneğin bir görüntünün) üzerine yerleştirilir. Merkez bileşeni ve çevresindeki piksellerin ağırlıklı toplamı hesaplanır ve hesaplanan ağırlıklı toplam merkez bileşeninin yerine geçirilir. Her çekirdek bir aktivasyon haritası üretir ve bu haritalar konvolüsyonel katmanın çıktısını oluşturmak için derinlik boyunca yığılır.

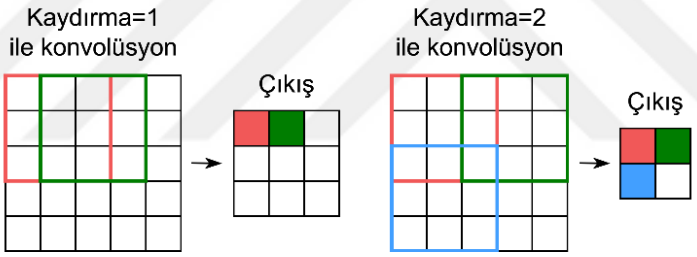


Şekil 4. Konvolüsyonel katmanın görsel temsili

Konvolüsyonel katmanlarda çıkış hacminin boyutunu kontrol eden üç parametre bulunur: Derinlik, Kaydırma (Stride) ve Zero-padding (Sıfır dolgusu). Bu parametreler çıkış hacminin nasıl boyutlandırılacağını ve düzenleneceğini belirler.

a) Derinlik: Çıkış hacminin derinliği konvolüsyon katmanındaki yerel bölgelere bağlanan nöron (yani filtre) sayısını belirler. Her bir filtre görüntünün kenarlar, köşeler ve desenler gibi farklı özelliklerini algılar ve bir aktivasyon haritası üretir. Derinlik ağın bu özellikleri öğrenme kapasitesini belirler.

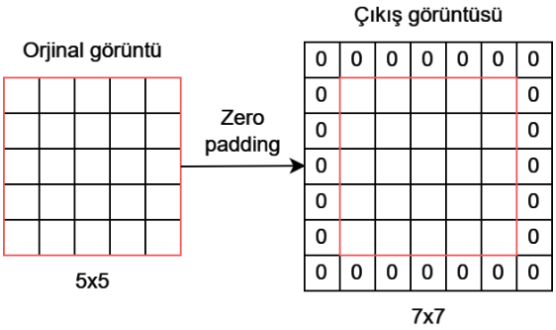
b) Kaydırma (Stride): Sinir ağı filtresinin giriş görüntüsü üzerinde hareket miktarını belirleyen parametredir. Filtre kaydırma değeri kadar piksel kaydırılır. Bu işlemin sonucunda filtre daha geniş bir alanda uygulanarak çıkış boyutu küçültülmüş olunur. Şekil 5'te kaydırma uygulaması ve denklem 1'de formülü verilmiştir. Bu denklemde I giriş görüntüsünün boyutunu, K filtre (kernel) boyutunu, S kaydırma miktarını ifade eder.



Şekil 5. Kaydırma uygulaması

$$\frac{\mathrm{I}-\mathrm{K}}{S}+1 \tag{1}$$

c) Zero-padding (Sıfır dolgusu): Görüntü üzerinde konvolüsyon işlemi yaparken temel görüntü boyutunu korumak amacıyla görüntünün kenarlarına padding uygulanması gerekir. Sıfır dolgusu (Zero-padding) uygulanarak giriş görüntüsü kenarlarından genişletilir ve çıkış hacminin giriş hacmiyle aynı boyutta olması sağlanır. Şekil 6'da Zero-padding uygulamasına örnek gösterilmiştir.



Şekil 6. Zero-padding uygulaması

3.3.2. Aktivasyon Katmanı

CNN'ler sinir ağlarına benzer şekilde aktivasyon fonksiyonlarını içeren bir katman kullanır. Bu katman genellikle aktivasyon katmanı olarak adlandırılır. Konvolüsyonel katmandan sonra kullanıldığı için topolojik şemalarda genellikle açıkça gösterilmez.

Aktivasyon fonksiyonları nöronun çıkışını giriş değerine göre belirler. CNN'ler için çeşitli aktivasyon fonksiyonları bulunmaktadır. Yerel çıktılar ürettikleri için ReLU aktivasyon fonksiyonları en yaygın kullanılanlardır. Bu girişin karmaşıklığını azaltarak ağın genelleme yapabilmesini veya veriye uyum sağlamasını mümkün kılar. Dolayısıyla aktivasyon fonksiyonları genellikle ReLU olarak adlandırılır. ReLU tanımı denklem 2'de verilmiştir.

$$ReLU(x) = max(x, 0) \tag{2}$$

ReLU giriş üzerinde basit bir eşikleme işlemi gerçekleştirir kolay ve hızlı bir şekilde uygulanabilir. Bu eşikleme işlemi yalnızca birkaç nöronun aynı anda aktive olmasına neden olur. ReLU'nun bu özellikleri genel hesaplama maliyetini etkili bir şekilde azaltmasına olanak tanır.

3.3.3. Havuzlama Katmanları

Havuzlama katmanları aktivasyon katmanının çıktısı üzerinde çalışır. Bu katmanın amacı özellik haritalarının boyutlarını kademeli olarak azaltmaktır. Özellik haritalarının boyutları azaltılırken önemli bilgiler korunur. Bu işlem ağdaki parametre ve hesaplama miktarını azaltmaya yardımcı olur ve aşırı öğrenmeyi (overfitting) engeller.

Havuzlama katmanı konvolüsyon katmanı tarafından üretilen özellik haritasının bir bölgesindeki özellikleri özetler. Bu özetlenmiş özellikler üzerinde daha ileri işlemler gerçekleştirilir. Böylece konvolüsyon katmanı tam olarak konumlandırılmış özellikler yerine özetlenmiş özelliklerle çalışır. Bu işlem modelin giriş görüntüsündeki özelliklerin konumundaki varyasyonlara karşı daha dayanıklı olmasını sağlar.

Havuzlama katmanlarında genellikle kullanılan mekansal havuzlama türleri maksimum havuzlama ve ortalama havuzlamadır. CNN mimarisinde mekansal boyutları azaltmak için genellikle maksimum havuzlama uygulanırken ağın son katmanında ise ortalama havuzlama kullanılır.

Maksimum Havuzlama: Filtre tarafından kapsanan özellik haritası bölgesindeki en yüksek değeri seçen bir havuzlama işlemdir. Bu işlem sonucunda maksimum havuzlama uygulanan özellik haritası önceki haritanın en belirgin özelliklerini içeren bir özellik haritası sağlar.

Ortalama Havuzlama: Filtre tarafından kapsanan özellik haritası bölgesindeki öğelerin ortalamasını hesaplar. Bu nedenle maksimum havuzlama belirli bir bölgede en belirgin özelliği seçerken ortalama havuzlama bölgedeki özelliklerin ortalamasını verir.

3.3.4. Düzleştirme Katmanı

Düzleştirme katmanı konvolüsyonel ve havuzlama katmanlarından gelen çok boyutlu veriyi tam bağlantılı katmanlarda işlenebilecek tek boyutlu bir vektöre dönüştürür. Bu işlem modelin özellik haritalarında öğrendiği bilgileri sınıflandırma katmanına iletebilmesini sağlar. Evrişimsel katmanlardan sonra elde edilen iki boyutlu veya üç boyutlu özellik haritaları düzleştirme katmanı sayesinde tek bir vektör haline gelir. Bu sayede modelin daha sonra gelen tam bağlantılı katmanlar ile sınıf tahminleri yapabilmesi için veriler uygun bir formata dönüştürülmüş olur.

3.3.5. Tam Bağlı Katmanlar

Tam bağlı katmanlar CNN'lerde özellik çıkarımını tamamlayan ve öğrenilen özellikleri çıktı sınıflarına bağlayan son adımdır. Tam bağlantılı katmanlar konvolüsyonel katmanlardan çok daha fazla parametre içerir ve bu yüzden genellikle modelin sonunda kullanılır. Bunun nedeni konvolüsyonel katmanların özellik haritalarını küçültmesi ve yerel

alıcı alanlar kullanarak daha yerelleştirilmiş sonuçlar üretmesidir. Tam bağlantılı katmanlar tüm girdiyi işlemek için çok fazla bağlantıya sahip olduğu için tüm ağırlıkları öğrenmeye çalışır. Fakat yerelleştirilmiş özellikleri öğrenmekte konvolüsyonel katmanlar kadar verimli olmadıklarından aynı sonuçları elde edebilmeleri için daha fazla eğitim süreci (epok) gerektirirler. Özetle konvolüsyonel katmanlar yerel bilgiye odaklanarak daha verimli sonuçlar üretebilirken tam bağlantılı katmanlar benzer sonuçlara ulaşmak için daha fazla eğitime ihtiyaç duyar.

Tam bağlantılı katman her bir nöronun bir boyutlu konvolüsyonlar yaptığı büyük bir konvolüsyonel katman olarak düşünülebilir. Önceki katmanlarda öğrenilen tüm bilgileri bir araya getirerek büyük desenleri öğrenir ve tüm girdi görüntüsünü doğru bir şekilde sınıflandırabilir. Sınıflandırma problemleri için son tam bağlantılı katmanın çıkış boyutu veri setindeki sınıf sayısına eşittir. Çıkışlar görüntünün hangi etiketle sınıflandırılacağına dair olasılıklar sağlar. En yüksek olasılığa sahip etiket sınıflandırma kararını belirler.

3.3.6. Softmax Katmanı

CNN modellerinde kullanılan softmax katmanı modelin son katmanında elde edilen çıktıyı her sınıf için olasılık değerine dönüştürür ve sınıflandırma problemlerinde modelin giriş verisinin hangi sınıfa ait olduğunu belirlemesine yardımcı olur. Softmax katmanı modelin çıkışında yer alan her bir değeri 0 ile 1 arasında normalize eder ve tüm sınıfların toplam olasılığının 1 olmasını sağlar. Bu şekilde modelin her sınıfa ait tahmin güveni olasılık olarak ifade edebilebilir.

Softmax katmanı her bir sınıfa ait skor değeri için denklem 3'teki formülü kullanır. Bu formülde K veri kümesindeki toplam sınıf sayısını, z ise bir önceki katmandan gelen çıkış değerlerini temsil eder. Her bir sınıf için z_i değeri kullanılarak bu sınıfa ait olasılık değeri hesaplanır. Bu sayede model tüm sınıflar arasında bir olasılık dağılımı oluşturur ve giriş verisinin en olası sınıfını yüksek bir olasılık değeri ile seçebilir.

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3}$$

3.4. Veri Artırma

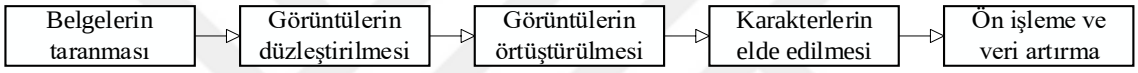
Veri artırma, makine öğrenmesi ve özellikle derin öğrenme alanında kullanılan bir yöntemdir. Bu yöntem mevcut veri kümesini genişletmek için çeşitli veri işleme teknikleri kullanarak yeni veri örnekleri üretir (van Dyk ve Meng, 2001). Daha fazla veri modelin öğrenme sürecinde daha çeşitli örneklerle karşılaşmasını ve böylece daha iyi performans göstermesini sağlar. Özellikle CNN, birçok sınıf içeren sınıflandırmalarda her sınıf için büyük miktarda eğitim verisine ihtiyaç duyar (Shorten ve Khoshgoftaar, 2019).

Görüntüler için veri artırma daha doğru bir model eğitmeye yardımcı olur. Küçük veri setlerinde aşırı öğrenme (overfitting) riski oldukça yüksektir. Bu durum modelin eğitim verisine fazla uyum sağlaması ve genel performansının test veya gerçek dünya verilerinde düşük olması anlamına gelir. Veri artırma daha genelleştirilmiş bir model geliştirerek aşırı öğrenme olasılığını azaltır (Sarah ve McGuinness, 2019). El yazısı karakterlerde bazı yaygın olarak kullanılan veri artırma yöntemler aşağıdaki gibidir:

- Döndürme (Rotasyon)
- Ölçekleme
- Kırpma
- Gürültü ekleme
- Erozyon ve genişletme
- Parlaklık ve kontrast ayarlama

4. YÖNTEM

CNN modelleri karakter tanıma gibi görevleri başarıyla yerine getirebilmek için genellikle büyük veri kümelerine ihtiyaç duyar. Büyük veri kümeleri, farklı yazım stilleri ve karakter varyasyonları gibi çeşitli durumları içerir. Böylece model farklı el yazısı stillerini ve karakter formlarını öğrenerek daha genel bir tanıma yeteneği kazanır. Türk alfabesindeki özel karakterleri içeren hazır veri setleri sınırlı olduğundan bu çalışmada kullanılan veri seti kendimiz tarafından hazırlanmıştır. CNN modellerinde kendi hazırladığımız veri setinin, özellikle Türkçe özel harflerin, performansını değerlendirmek amaçlanmıştır. Bu veri setinin hazırlanma aşamaları Şekil 7'de gösterilmiştir.

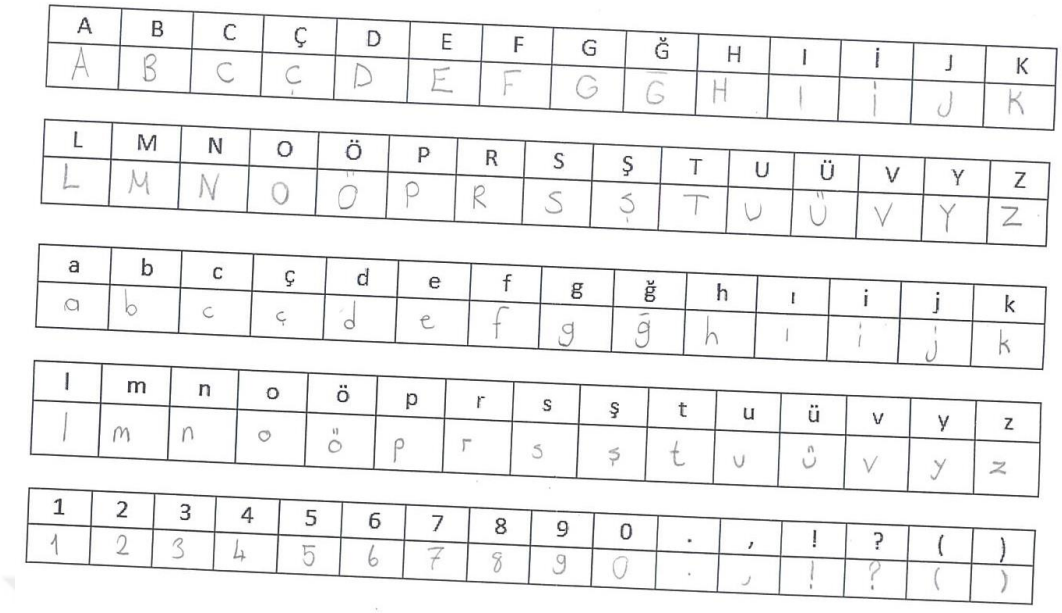


Şekil 7. Veri setinin hazırlanma aşamaları

4.1. Belgelerin Taranması

Veri setinin hazırlanması için kullanılan taban belge, A'dan Z'ye kadar büyük ve küçük Türkçe harfler ve 0-9 arası rakamlar olmak üzere toplamda 68 farklı karakter içermektedir. Bu belgelerde üst kısımlarda karakterin dijital yazısı, alt taraflarda ise bu karakterin el yazısıyla yazılması için boş kare şekiller bulunmaktadır. Bu belgeler bir kurumda çalışan gönüllü 400 kişi tarafından farklı yazım stilleri ile doldurulmuştur ve ardından taratılmıştır.

Belgelerin taratılması esnasında belgelerin düzgün yerleştirilememesi ve donanımsal sorunlar gibi nedenlerden dolayı taratılan görüntüler içerisinden bir çoğu istenmeyen şekilde eğik olarak dijital ortama aktarılmıştır. Görüntülerin eğik bir şekilde taratılmış olması karakterlerin doğru bir şekilde elde edilmesini zorlaştırmaktadır. Aynı zamanda karakterlerin asıl yazılışı yerine eğik bir şekilde elde edilmesi istenmemektedir. Bundan dolayı taratılan görüntüler ilk aşamada düzleştirilmiştir. Şekil 8'de taratılan görüntülerden birinin eğikliği örnek olarak gösterilmiştir.

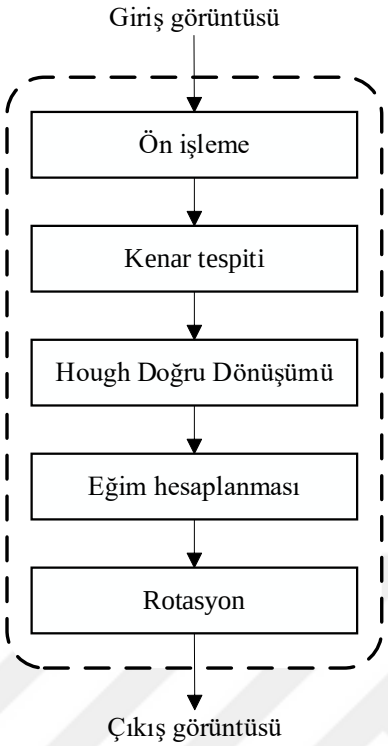


Şekil 8. Taratılan görüntü örneği

4.2. Görüntülerin Düzleştirilmesi

Taratılan görüntülerden el yazısı karakterlerin kolayca elde edilebilmesi için görüntülerin düzleştirilmesi gereklidir. Bu sayede aynı zamanda karakterlerin temel görüntüsü korunarak eğikliği giderilir. Az sayıda görüntüde bu işlem farklı yollarla kolaylıkla yapılabilir fakat 400 tane her bir görüntü için farklı açılarda eğikliğe sahip görüntülerde bu işlem zorlayıcıdır.

Bu sorunun üstesinden gelmek için Hough Doğru Dönüşümü kullanılmıştır. Hough Doğru Dönüşümü ile görüntülerdeki çizgiler tespit edilir, tespit edilen çizgiler arasından en uzunu seçilir ve bu çizgi referans alınarak görüntüde düz yani yatay olacak şekilde görüntüye rotasyon yaptırılır. Görüntülerin düzleştirilmesi aşamasında yapılan işlemler Şekil 9'da gösterilmiştir.



Şekil 9. Görüntülerin düzleştirilmesi aşamaları

4.2.1. Ön İşleme

Renkli (RGB) bir görüntüde her pikselin kırmızı, yeşil ve mavi bileşenleri [0, 255] aralığında değişmektedir. Hough Doğru Dönüşümü gibi algoritmaların daha verimli çalışabilmesi ve kenar tespitinin daha hassas yapılabilmesi amacıyla görüntülerin ikili formata dönüştürülmesi gerekmektedir. Bu amaçla işlenecek giriş görüntüleri öncelikle gri tonlamaya dönüştürülür ve ardından iki değere indirgenir. Bu sayede çoğu kenar tespit algoritması için gereksiz olan renk bilgisinden kurtulmuş olunur ve görüntüler iki değere indirgenerek her piksel sadece 0 (siyah) veya 1 (beyaz) olur.

RGB bir görüntüyü gri tonlamalı bir görüntüye dönüştürme süreci her pikselin renkli bileşenlerinin belirli ağırlıklarla birleştirilmesiyle gerçekleşir. Bu sayede görüntülerdeki renk farkları değil yalnızca parlaklık farkları ortaya çıkarılır. RGB bir görüntüyü gri tonlamaya çevirme işlemi denklem 4'de ifade edilmiştir. Bu denklemde verilen ağırlık değerleri insan gözünün renkleri algılama biçimine uygun olarak belirlenmiştir.

$$Gri\ Tonlama = (R \times 0.299) + (G \times 0.587) + (B \times 0.114) \quad (4)$$

Gri tonlama ve ikili dönüşüm işlemlerinin ardından giriş görüntüleri gaussian filtresi ile bulanıklaştırılarak gürültü azaltılır. Burada amaç görüntülerdeki rastgele piksellerin neden olduğu istenmeyen varyasyonları ve bozulmaları (gürültüyü) azaltmaktır. Bu işlem kenar tespiti ve diğer görüntü işleme işlemlerinin daha doğru ve güvenilir sonuçlar üretmesini sağlar. Gaussian filtresi denklem 5 ile tanımlanır. Burada (x, y) piksel konumunu ve σ Gaussian dağılımının standart sapmasını temsil eder.

Bu bölümde anlatılan ön işleme yöntemleri algoritmaların çalışma aşamasında kullanılır ve nihai görüntüde bir değişiklik yapmaz. Bu ön işleme aşamaları daha sonra bahsedilecek uygulamalarda da kullanılacaktır.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

4.2.2. Kenar Tespiti

Görüntülerdeki belirgin kenarları bulmak için Canny kenar tespiti kullanılır. Canny kenar tespiti görüntülerdeki geniş bir kenar yelpazesini tespit etmek için çok aşamalı bir algoritma kullanan bir kenar algılama operatörüdür. Bu algoritma kenarları tespit ederek bir kenar haritası oluşturur. Kenar haritası görüntüdeki farklı bölgelerdeki yoğunluk değişimlerini temsil eder. Elde edilen kenar haritası daha sonra Hough Doğru Dönüşümü uygulaması için kullanılır ve görüntüdeki düz çizgilerin bir kümesi çıkarılır. Canny kenar tespiti birkaç adımı içerir:

Gradyan Hesaplama: Görüntülerdeki gradyanı hesaplamak için Sobel operatörleri kullanılır. Bu işlem x ve y yönlerinde gradyanları (veya kenarların eğimlerini) verir. Sobel operatörler denklem 6 ve 7'da verilmiştir. Uygulamada kenar tespiti için kullanılan Sobel operatörünün pencere boyutu 3 olarak seçilmiştir. Bu, 3x3 boyutunda bir pencere kullanılarak gradyanların hesaplanacağını ifade eder.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{6}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{7}$$

Sobel filtreleriyle elde edilen yatay gradyan G_x ve düşey gradyan G_y kullanılarak her piksel için gradyan büyüklüğü denklem 8 ve gradyan yönü denklem 9 ile hesaplanır. Bu hesaplamalar kenarların ne kadar belirgin olduğunu ve hangi yönde olduğunu belirler. Denklem 8'de verilen θ gradyan yönünü (açısını) temsil eder.

$$Gradyan\ Büyüklüğü = \sqrt{G_x^2 + G_y^2} \tag{8}$$

$$\theta = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{9}$$

Maksimum Olmayanların Bastırılması: Bu adım kenarları keskinleştirmek ve gereksiz zayıf kenarları ortadan kaldırarak net bir kenar haritası oluşturmayı sağlar. Bu işlem gradyan yönünde komşu piksellerle karşılaştırılarak yapılır. Böylece sadece en yüksek gradyan değeri olan pikseller kenar olarak kabul edilir ve diğerleri yok sayılır.

İkili Eşikleme: İkili eşikleme aşamasında düşük ve yüksek eşik değerleri kullanılarak önemli kenarlar seçilir ve zayıf olanlar elenir. Yüksek eşikten büyük olan pikseller kenar olarak kabul edilirken düşük eşikten küçük olan pikseller kenar olarak işaretlenmez.

Histerezis Eşikleme: Bu adımda düşük ve yüksek eşik değerleri arasındaki pikseller bağlanarak keskin kenarlar oluşturulur. Yüksek eşik değerinden büyük olan pikseller kenar olarak işaretlenir ve düşük eşik değerinin altındaki pikseller yalnızca yüksek eşik değerine bağlı olarak kenar olarak kabul edilir. Bu işlem kenarların daha belirgin ve keskin olmasını sağlar. Kenar tespiti için kullanılan parametrelerde düşük eşik değeri 30, yüksek eşik değeri 100 olarak seçilmiştir.

4.2.3. Hough Doğru Dönüşümü

Dijital görüntü analizlerinde karşılaşılan önemli zorluklardan biri şekil tespitidir. Görüntülerdeki düz çizgiler, elipsler ve daireler gibi geometrik desenlerin tespiti çok sayıda hesaplama ve işleme ihtiyaç duyar. Düz çizgilerin tespiti için kullanılan yöntemlerden biri olan Hough Doğru Dönüşümü bu sorunun çözümüne yönelik bir tekniktir (Mukhopadhyay ve Chaudhuri, 2015).

$$y = mx + c \tag{10}$$

Doğrular genellikle denklem 10'da verildiği gibi parametrik hale getirilir. Bu denklemde, m doğrusal eğimi, c ise doğrunun y-eksenini kestiği noktayı temsil eder. Ancak dikey doğrular için m'nin sonsuz bir değere ulaşması sorun oluşturur. Bu nedenle doğrunun dik bir segmentinin oluşturulması gereklidir.

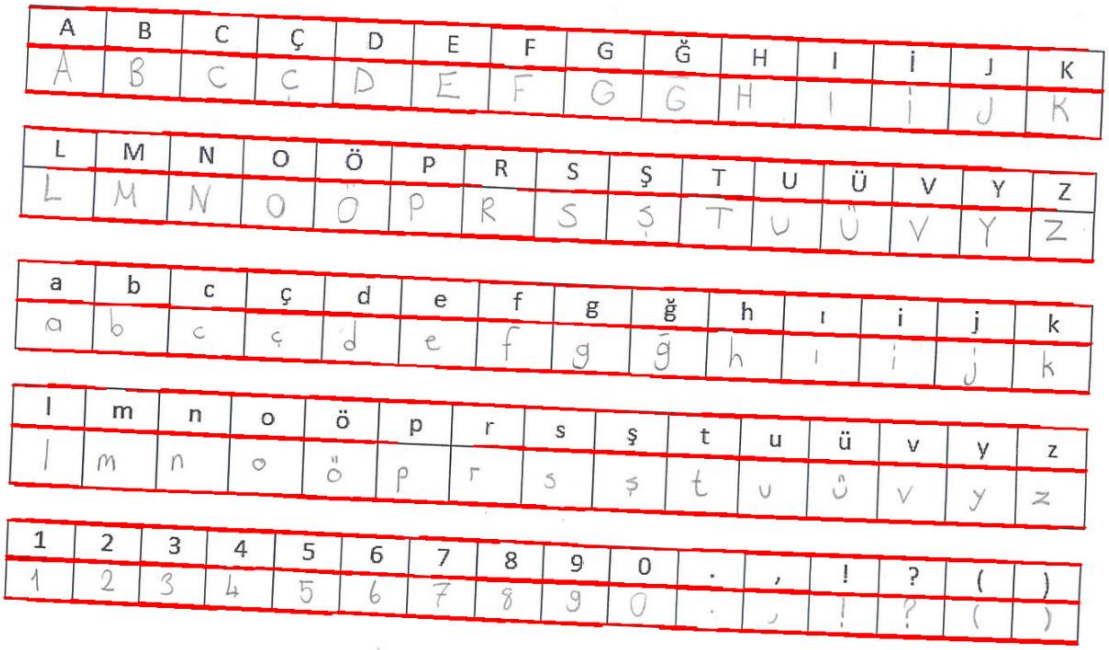
$$\rho = x \cdot cos(\theta) + y \cdot sin(\theta) \tag{11}$$

Hough Doğru Dönüşümünde doğrular denklem 11'de verildiği gibi parametrize edilir. Burada doğrunun uzunluğu ρ ve x-eksenine göre eğimi θ ile ifade edilir. Hough Doğru Dönüşümü farklı ρ ve θ değerlerini içeren bir parametre uzayı histogramı oluşturur. Her bir ρ ve θ değeri için giriş görüntüsündeki sıfır olmayan pikseller değerlendirilir ve bu piksellere karşılık gelen histogram hücresi artırılır. Sıfır olmayan her piksel potansiyel doğrular için bir oy kullanır. Oluşan histogramdaki yerel maksimum noktalar en olası doğruları işaret eder ve bu noktalar görüntüdeki düz çizgileri belirler.

Uygulamada taranan görüntüler üzerinde çizgilerin en iyi şekilde tespit edilebilmesi için Hough Doğru Dönüşümü'nde kullanılan parametreler aşağıda verilmiştir.

- ρ (rho)=0,4: Çizgi tespiti olarak kullanılan çözünürlük birimidir. Çizgilerin tespit edilme hassasiyetini etkiler.
- Θ (theta): Radyan cinsinden açısal çözünürlük. Uygulamada 1 derece seçilmiştir (np.pi/180).
- threshold=30: Bir çizginin tespit edilmesi için gereken minimum oy sayısı. Bu eşik değerinin altında kalan çizgiler dikkate alınmaz.
- minLineLength=35: Çizgi uzunluğu için minimum uzunluk. Bu uzunluğun altında kalan çizgiler tespit edilmez.
- maxLineGap=10: Çizgiler arasında maksimum boşluk. Bu boşluktan daha büyük boşluklar içeren çizgiler birleştirilmez.

Yukarıda verilen parametreler ile birlikte taratılan belgeler üzerindeki dikey çizgiler gibi istenmeyen çizgilerin tespit edilmemesi için ayrıca eğim açısı filtresi uygulanmıştır. Taranan görüntülerde Hough Doğru Dönüşümü ile tespit edilen çizgilerin bir örneği Şekil 10'da gösterilmiştir.



Şekil 10. Hough Doğru Dönüşümü uygulamasına örnek görüntü

4.2.4. Eğim Hesaplanması ve Rotasyon

Bu aşamada amaç Hough Dönüşümü sonucunda görüntüler üzerinde tespit edilen düz çizgiler arasından en uzun olanı referans alarak bu çizginin görüntü üzerinde yatay yani düz olacak şekilde görüntülerin rotasyonunu yapmaktır.

İlk olarak her bir çizginin uzunluğu hesaplanır ve bu uzunluklara göre en uzun çizgi seçilir. Tespit edilen çizgilerin uç noktaları $(x_1, y_1)\ ve\ (x_2, y_2)$ olduğunu varsayarsak çizginin uzunluğu L denklem 12'e göre hesaplanır. Bu formül iki nokta arasındaki Öklidyen mesafeyi verir ve çizginin uzunluğunu ifade eder. Her bir çizgi için bu uzunluk hesaplanır ve en uzun çizgi seçilir.

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{12}$$

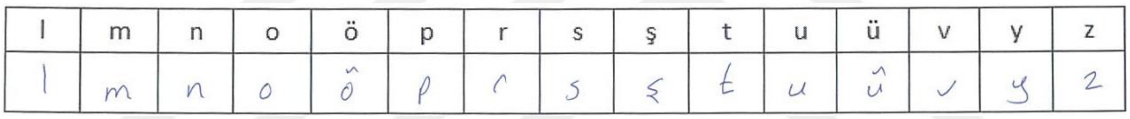
En uzun çizginin açısı Hough Dönüşümü'nden alınan θ değerine göre hesaplanır. θ çizginin x ekseniyle yaptığı açıdır ve bu açı radyan cinsindendir. Radyanı dereceye çevirmek için denklem 13 kullanılır. Bu işlem çizginin eğim açısını verir. Hesaplanan açıya göre görüntü döndürülür. Çizginin yatay hale getirilmesi için (yani açı 0 derece) görüntü düzeltilir.

$$Açı\ (derece) = \theta.\left(\frac{180}{\pi}\right) \quad (13)$$

Görüntülerin düzleştirilmesi aşaması sonucunda taratılan görüntülerin, dolayısıyla karakterlerin eğiklikleri düzleştirilmiştir. Bu işlemler taranan görüntülerden karakterlerin elde edilmesini kolaylaştırır.

4.3. Görüntülerin Örtüştürülmesi

Bu aşamaya gelene kadar görüntülerin eğikliği düzeltilmiştir. Fakat görüntüler hala aynı hizada değildir. Bu aşamada ise görüntüler üst üste hizalanarak karakterlerin elde edilmesinin kolaylaştırılması hedeflenmektedir. Bu işlemi bütün görüntüde uygulamak zorlayıcı olacaktır. Fakat tüm görüntülerden Şekil 11'de görüldüğü gibi karakterlerin yazılı olduğu her bir satır kırpılıp bu yeni görüntülerin üst üste hizalanması sağlanabilir.



Şekil 11. l-z arası harfleri içeren satır

Bu işlem tüm görüntüde belirli bir alandaki en büyük dikdörtgen şeklin tespit edilerek kırpılması ile yapılır. Bu işlemin adımları aşağıda verilmiştir. Bu adımlardan önce görüntülere daha önce anlatılan ön işleme yöntemleri uygulanır.

- İlk adımda görüntüdeki kenarları belirlemek için Canny kenar tespiti uygulanır. Bu yöntem kırpılması istenen satır aralığına göre kenarları algılar. Tespit edilen kenarlar konturların bulunmasına olanak tanır.
- Kenar tespitinden elde edilen ikili görüntü üzerinde konturlar bulunur. Konturlar her nesneyi sınırlayan bir çerçeve oluşturur.
- Her kontur için bir dikdörtgen sınır belirlenir. Bu dikdörtgenlerin alanları hesaplanır ve en büyük alanlı dikdörtgen seçilir. Seçilen en büyük dikdörtgen görüntüden Şekil 11'de görüldüğü gibi kırpılır.

Taratılmış görüntülerin hepsinin aynı taban belgeden oluşturulduğu göz önünde bulundurulursa bu görüntüler düzleştirilip aynı alandan kırpıldığında kırpılan görüntüler üst üste hizalanmış olur. Örneğin görüntüler düzleştirildikten sonra Şekil 11'de verilen örnekte

olduğu gibi l-z arası harfleri içeren satırlar tüm görüntülerden kırpıldığında bu satırların görüntülerinin hepsi artık üst üste hizalanmış olur. Bu işlem karakterleri içeren tüm satırlar için uygulanmıştır ve bu aşamadan sonra görüntülerden karakterlerin kolayca elde edilebilmesi mümkün olmuştur.

4.4. Karakterlerin Elde Edilmesi

Karakterlerin üst üste yerleştirilmiş görüntülerden elde edilebilmesi için öncelikle karakterleri içeren kare şekillerin piksel koordinatlarına göre kırpma yapılması düşünülmüştür. Örneğin içerisinde “m” harfini bulunduran kare şeklin koordinatları tüm görüntülerde yaklaşık x ekseninde 90’dan 180’e, y ekseninde ise 57’den 126’ya kadar olan alanda bulunur. Bu alan tüm görüntülerden kırpılarak m harfleri elde edilebilir. Fakat bu yöntemde bazı problemlerle karşılaşılmıştır. Bazı boyutu büyük olan karakterlerin içerisinde bulunduğu kare şeklin kenarlarına kadar uzanmasından dolayı kırpılma esnasında karakterin içeriği de kırpılmıştır. Örneğin "E" harfi kare şeklin sınırlarına kadar uzandığında alt kısmının kesilmesi durumunda "F" harfine benzeyen bir sonuç ortaya çıkabilir ve bu modelin performansını olumsuz yönde etkileyebilir. Bir başka problem ise bazı kırpılan görüntülerin en kenar bölgesinde kare şeklin çizgisi tespit edilmiştir.

Bu problemlerin üstesinden gelmek ve karakterleri en iyi şekilde elde edebilmek için harfleri içeren satırlarda el yazısı karakterlerin içerisinde bulunduğu kare şekillerin tespit edilip görüntülerden kırpılması ve her karakterin görüntülerinin karakterin ismiyle oluşturulan klasörlere kaydedilmesi düşünülmüştür. Bu işlem aşağıda verildiği şekilde gerçekleştirilir. Bu adımlardan önce görüntülere yine ön işleme uygulanmıştır.

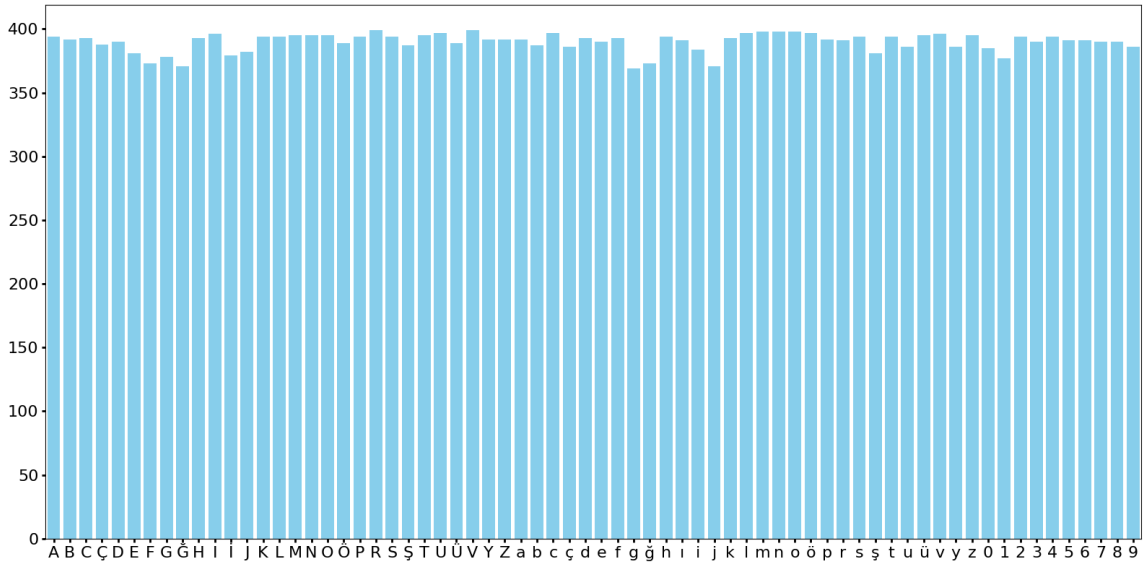
Kenar Tespiti: Canny kenar tespiti algoritması ile görüntüdeki kenarlar bulunur. Bu kontur bulma işlemi için temel bir adımdır. Görüntüdeki nesnelerin sınırlarını bulmak için kenarların belirlenmesi gerekir.

Kontur Bulma: Bu adımda kenarların oluşturduğu konturlar bulunur. Bu konturlar görüntüdeki nesnelerin sınırlarını temsil eder. Konturlar bir şeklin çevresini tanımlayan bir dizi noktadan oluşur. Kare veya dikdörtgen nesneleri bulmak için konturlar üzerinden şekil analizine başlanır.

Şekil Yaklaştırma: Bulunan konturların her biri basitleştirilir. Bu işlem bir konturu daha az sayıda köşe ile temsil eder. Konturun dört köşesi (4 nokta) varsa bunun bir kare veya dikdörtgen olduğu varsayılır.

Alan ve Oran Filtreleme: Konturun genişliği ile yüksekliği arasında oran hesaplanır ve belirli bir aralıkta (0.5 < en-boy oranı < 2.0) olup olmadığını kontrol edilir. Bu aralık el yazılarını içerisinde barındıran kare şekillere göre belirlenmiştir. Eğer bir konturun alanı belirtilen sınırlar içerisinde ise kare olarak kabul edilir.

Bu adımların sonucunda belirlenen alan içinde kalan kareler tespit edilir ve kırpılarak kaydedilir. Bu işlem tüm karakterler için uygulanarak veri seti oluşturulmuştur. Veri seti oluşturulduktan sonra titizlikle incelenmiş ve kontrol edilmiştir. Bu incelemeler sırasında bazı görüntülerde el yazısı hataları, karakterlerin kare şeklin dışına taşması gibi sorunlar tespit edilmiş ve bu tür hatalı görüntüler veri setinden çıkarılmıştır. Veri setinin bu şekilde düzeltilmesi modelin eğitimi sırasında hatalı veri girişinden kaynaklanabilecek sorunların önüne geçmek amacıyla yapılmıştır. Bu işlemler sonucunda elde edilen veri setindeki 26509 adet karakterin dağılımının histogramı Şekil 12'de gösterilmiştir.



Şekil 12. Karakter dağılım histogramı

4.5. Veri Ön İşleme

Veri ön işleme yöntemleri veri kalitesini artırmak ve CNN modelinin performansını iyileştirmek için kritik bir adımdır. El yazısı karakterleri gibi görsel verilerle çalışırken veri ön işleme daha temiz, daha tutarlı ve modelin anlamlandırabileceği bir veri seti elde etmeyi sağlar.

Sinir ağı modellleri, özellikle CNN gibi yapılarda giriş görüntülerinin belirli sabit bir boyutta olması gerekir. El yazısı karakterler taratılan görüntülerden elde edilirken kare

şekiller tespit edilip kırpılmıştır. Görüntülerdeki kare şekillerin boyutları yakın olmasına rağmen kırpılan görüntülerin her biri aynı boyutta değildir. Veri kümesindeki görüntüler farklı boyutlarda olduğundan bu görüntülerin CNN modelinin beklediği sabit boyuta getirmek gerekir. Ancak bu işlemi yaparken görüntülerin kalitesini bozmadan ya da asıl şekillerini değiştirmeden boyutlandırmak önemlidir. Bu sayede modelin tutarlı bir giriş boyutu elde etmesi ve öğrenme sürecinin daha verimli hale gelmesi sağlanır.

Ayrıca veri setindeki görüntülerde el yazısının bulunduğu konumun görüntülerin tam ortasında yer alması modelin performansı için daha iyi bir durumdur. El yazısı karakterlerin görüntülerin merkezinde yer alması modelin ilgili özellikleri (karakterin şekli, hatların kalınlığı, boşluklar vb.) daha iyi öğrenmesine yardımcı olur. Veri setindeki karakterlerin görüntülerde konumları değişkendir. Örneğin bazıları ortada, bazıları ise köşelerde bulunur. Bu durum modelin öğrenmesi gereken daha fazla varyasyona sebep olur ve modelin kararsız hale gelmesine veya belirli bir karakter üzerinde iyi bir performans göstermemesi ile sonuçlanabilir.

Bu nedenlerden dolayı karakterlerin hem görüntülerde merkeze hizalanması hem de görüntülerin CNN modelinin giriş boyutu olan 64x64 piksel boyutuna getirilmesi için boyut normalizasyonu yapılır. Bu amaçla veri setindeki görüntülerde el yazısı karakterler sınır kutuları tespit edilerek kırpılır ve ardından CNN modelinin giriş boyutu olan 64x64 piksel boyutuna kadar beyaz rankle padding uygulanır. Bu işlem karakterlerin kalitesini bozmadan ve gerçek şekillerini koruyarak boyutlandırmayı ve karakterlerin görüntülerin merkezine hizalanmasını sağlar.

4.5.1. Karakterlerin Sınır Kutusu Tespiti

Karakterlerin sınır kutularının tespit edilme işlemi görüntülerde el yazısı karakterlerin tespit edilip görüntüden kırpılma işlemidir. Sınır kutuları karakterlerin etrafında çizilen dikdörtgenlerdir. Karakterlerin görüntüleri beyaz arka plan üzerinde olmasından dolayı karakterler kolayca tespit edilip görüntülerden kırpılabilir. Bu yüzden karakterlerin sınır kutularının tespiti karakterler taratılan görüntülerden elde edildikten sonra uygulanmıştır. Bu işlem karakterlerin elde edilmesi adımındaki kare şekillerin tespit edilerek kırpılması yöntemine benzer. Aşağıda bu işlemin adımları sırasıyla verilmiştir.

Morfolojik Genişletme: İkili görüntüdeki beyaz alanları büyütmek ve aralarındaki boşlukları doldurmak için kullanılır. Bu işlem karakterlerin birbirine yakın veya kesik

görünmesini engeller. Böylece karakterler daha belirgin hale gelir ve kontur tespit süreci daha başarılı olur.

Morfolojik Kapama: Görüntülerdeki küçük boşlukları kapatmak için uygulanır. Bu işlem daha önce yapılan genişletme adımından sonra oluşabilecek küçük deliklerin doldurulmasını sağlar. Bu sayede karakterlerin tamamı bir bütün olarak algılanır ve tespit işlemleri sırasında yanlış pozitif veya negatif sonuçların önüne geçilir.

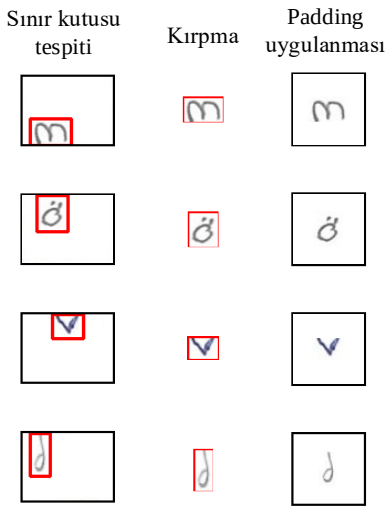
Kontur Bulma: İşlenmiş görüntülerdeki konturlar tespit edilerek karakterlerin sınırları belirlenir. Kontur bulma karakterleri ayırt etmek için kritik bir adımdır.

Alan Filtreleme: Bu adımda küçük alanlara sahip konturlar göz ardı edilir. Minimum alan değeri 50 piksel olarak belirlenmiştir. Küçük konturlar genellikle gürültü veya hata kaynaklarıdır ve gerçek karakterleri temsil etmeyebilir. Bu nedenle yalnızca belirli bir büyüklüğe ulaşan konturlar dikkate alınarak modelin doğruluğu artırılmaya çalışılır.

En Büyük Konturu Bulma: Filtrelenmiş konturlar arasından en büyük konturun seçilmesi genellikle görüntüdeki en belirgin karakterin tespit edilmesini sağlar. Bu konturun etrafında bir dikdörtgen tanımlanır. Bu işlem karakterin kesilerek çıkarılması için gereken sınırları belirler.

4.5.2. Padding Uygulanması

Sınır kutuları tespit edilerek kırpılan karakterler CNN modelinin giriş boyutu olan 64x64 piksel boyutuna kadar beyaz padding uygulanarak hedef boyuta ulaştırılır. Aşağıda bazı görüntülerde karakterlerin sınır kutularının tespit edilme işlemi ve ardından padding uygulanmasına örnek verilmiştir. Şekil 13’de görüntünün köşelerinde bulunan el yazısı karakterler sınır kutusu tespiti ile tespit edilerek kırpılmış, ardından 64x64 hedef boyutuna kadar beyaz renk padding uygulanarak görüntünün merkezine hizalanmıştır.



Şekil 13. Sınır kutusu tespiti ve padding uygulanması

4.6. Veri Artırma

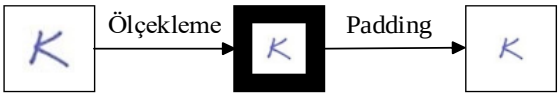
Veri artırma özellikle derin öğrenme ve görüntü işleme görevlerinde oldukça önemli bir tekniktir. CNN modeli yalnızca temel veri setindeki karakterleri öğrenirse gerçek dünyadaki varyasyonları tanımakta zorlanabilir. Veri artırma yöntemleri modelin daha çeşitli verilerle eğitilmesini sağlar ve aşırı öğrenme riskini azaltır.

4.6.1. Ölçekleme

Ölçekleme el yazısı karakterlerde sıklıkla kullanılan veri artırma yöntemlerinden biridir. Karakterlerin boyutları 0,7 ile 1,2 arasında rastgele bir ölçek faktörü kullanılarak değiştirilir ve yeni varyasyonlar üretilir. Karakterlerin farklı boyutlarda görülebilmesi modelin boyut varyasyonlarına karşı daha iyi genelleme yapmasını sağlar. Bu gerçek dünyadaki farklı boyutlarda yazılmış karakterleri tanıma yeteneğini artırır.

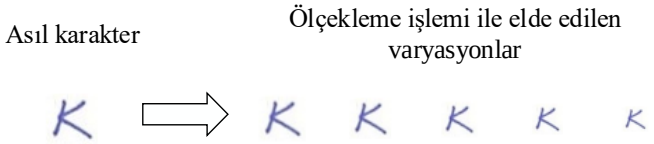
Karakterler üzerinde ölçekleme işlemi uygulanırken bazı noktalara dikkat edilmiştir. Büyük ve küçük halleri şekil olarak aynı olan harfler (C ve c, Ç ve ç, O ve o, Ö ve ö, S ve s, Ş ve ş, U ve u, Ü ve ü, Z ve z) sadece boyut farkıyla ayırt edilirler. Bu durum özellikle el yazısında ve karakter tanıma sistemlerinde CNN gibi modellerde sorun yaratabilir.

Ölçekleme işlemi görüntünün sınırları dışında kalan alanlarda siyah alanlar oluşturur. Bu istenmeyen bir durumdur. Küçültülen görüntülerde siyah alanları engellemek için giriş boyutuna uyacak şekilde görüntüye beyaz renk ile padding uygulanır. Bu işlem Şekil 14'de gösterilmiştir.



Şekil 14. Ölçekleme ve padding işlemi

Bu işlemler sonucunda yeni oluşan varyasyonlar veri setine eklenir. Veri setindeki bir karakter için ölçekleme işlemine örnek Şekil 15'te gösterilmiştir.

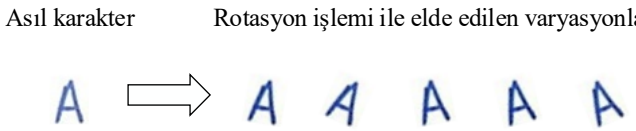


Şekil 15. Ölçekleme işlemine örnek

4.6.2. Görüntüleri Döndürme (Rotasyon)

HCR sistemlerinde görüntüleri döndürme (rotasyon) yaygın olarak kullanılan veri artırma tekniklerinden biridir. Veri setindeki her bir görüntü belirli bir rotasyon aralığında rastgele döndürülerek yeni varyasyonlar üretilir ve bu varyasyonlar veri setine kaydedilir. Bu işlem modelin farklı açılardaki varyasyonları öğrenmesine olanak tanır.

Rotasyon açısı karakterlerin tanınabilirliğini koruma amacıyla 0-30 derece gibi makul bir aralıkta tutulmuştur. Bunun nedeni, örneğin 6 sayısı 180 derece döndürüldüğünde 9 ile karıştırılabilir. Bu tür hataları önlemek için rotasyon açısı dikkatle seçilmiştir. Büyük açılardan kaçınılarak karakterlerin stillerini ve yazı biçimlerini bozmadan modelin farklı değişkenlikleri öğrenmesi sağlanır. Rotasyon işlemine örnek Şekil 16'da gösterilmiştir.



Şekil 16. Rotasyon işlemine örnek

4.7. Eğitim Aşaması

Eğitim aşamasında başlangıçta veri setlerinde ön işleme ve veri artırma yöntemlerinin CNN modellerinin performansına olan etkisi incelenecektir. Bu amaçla karakterlerin ham hali olan veri seti, ön işleme uygulanmış veri seti ve hem ön işleme hem de veri artırma

yöntemleri uygulanmış veri seti ile yapılan eğitim sonuçları karşılaştırılacaktır. Bu veri setlerinin en iyi şekilde karşılaştırılabilmesi için tüm veri setleri aynı CNN modeli ile eğitilmiştir. Bu CNN modeli daha sonra "CNN Modelleri" başlığı altında verilecek olan CNN5 modeli seçilmiştir.

Tüm veri setleri, eğitim, doğrulama ve test setleri olmak üzere üçe ayrılmıştır. Eğitim verisi modelin öğrenmesi için kullanılır. Model bu veri setinden özellikleri öğrenir ve ağırlıklarını güncelleyerek karakter tanıma görevini yerine getirmeyi öğrenir. Doğrulama verisi eğitim sırasında modelin genel performansını kontrol etmek için kullanılır ve modelin eğitim verisinde aşırı uyum sağlamasını (overfitting) önlemeye yardımcı olur.

4.7.1. Modelin Derlenmesi ve Eğitim Parametreleri

Modellerin eğitiminde karakter sınıflarını öğrenmek için Adam optimizasyon algoritması kullanılmıştır. Bu algoritma öğrenme sürecini hızlandıran ve verimli hale getiren bir optimizasyon yöntemidir. Düşük bir öğrenme oranı modelin kararlı bir şekilde öğrenmesine olanak tanır ancak daha uzun eğitim süresi gerektirir (Kingma ve Ba, 2015).

Hata ölçümünde ise kayıp fonksiyonu olarak Kategorik Çapraz Entropi kullanılmıştır. Bu fonksiyon çok sınıflı sınıflandırma problemlerinde her sınıf için üretilen olasılık tahminlerinin gerçek etiketlerle ne kadar uyuştuğunu hesaplar (Heaton, 2017). Bu nedenle çok sayıda harf ve rakam sınıfı içeren el yazısı karakter tanıma görevinde etkili bir kayıp fonksiyonudur.

Eğitim verileri büyük ve küçük Türkçe harfler ile rakamları içeren etiketlenmiş karakterler kümesinden alınır. Bu karakter kümeleri etiketlenmiş karakterlerin klasörler içerisinde görüntülerini içerir ve 68 sınıftan oluşmaktadır. Bu sınıflar aşağıda verilmiştir:

```
ABCÇDEFGĞHIİJKLMNOÖPRSŞTUÜVYZ
abcçdefgğhıijklmnoöprsştuüvyz
0123456789123456789
```

Tüm veri setleri; %80 eğitim, %10 doğrulama ve %10 test verileri olarak ayrılır. Aşağıda tüm modellerin eğitiminde ortak kullanılan parametreler verilmiştir.

- Epok sayısı = 50
- Batch boyutu = 32
- Öğrenme oranı = 0.0001

4.7.2. Veri Seti-1 ile Eğitim

Veri seti-1, karakterlerin ilk elde edildiği (veri setinin ham hali olan) veri setidir ve 26509 adet etiketlenmiş karakterden oluşmaktadır. Bu karakterlerin dağılım histogramı daha önce "Karakterlerin Elde Edilmesi" başlığı altında Şekil 12'de verilmiştir. Toplam eğitim veri sayısı 26509 olan veri seti-1 için batch boyutu 32'dir. Bu durumda her epok başına adım sayısı denklem 14'te verildiği gibi hesaplanır. Bu hesaplamaya göre model her biri 828 adım içeren 50 epok süresince eğitilir.

$$\text{Epok başına adım sayısı } = \frac{26509}{32} \approx 828 \tag{14}$$

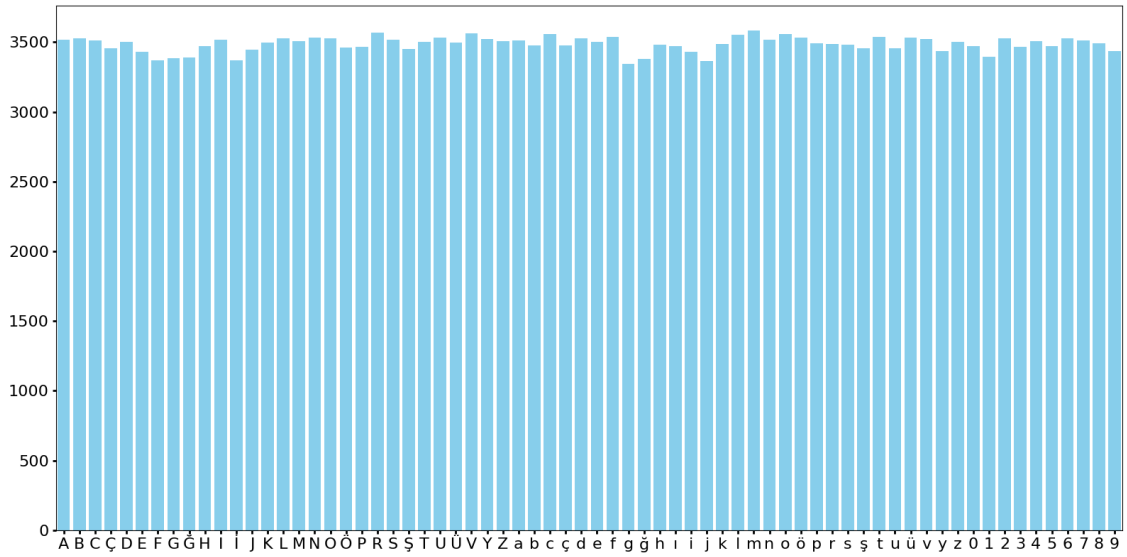
4.7.3. Veri Seti-2 ile Eğitim

Veri seti-2, ham veri setine (veri seti-1) ön işleme uygulanmış veri setidir. Ön işleme aşamasında karakterler görüntülerin merkezine hizalanmış ve modelin beklediği standart boyuta (64x64) padding uygulanarak getirilmiştir. Toplam eğitim veri sayısı veri seti-1 ile aynıdır ve her biri 828 adım içeren 50 epok süresince eğitilir.

4.7.4. Veri Seti-3 ile Eğitim

Veri seti-3, ön işleme uygulanmış veri setine (veri seti-2) rotasyon ve ölçekleme gibi veri artırma yöntemleri uygulanarak elde edilmiştir. 237089 adet etiketlenmiş karakterden oluşmaktadır. Bu karakterlerin dağılım histogramı Şekil 17'de gösterilmiştir. Toplam eğitim veri sayısı 237089 olan veri seti-3 için batch boyutu 32'dir. Bu durumda her epok başına adım sayısı denklem 15'te verildiği gibi hesaplanır. Bu hesaplamaya göre model her biri 7409 adım içeren 50 epok süresince eğitilir.

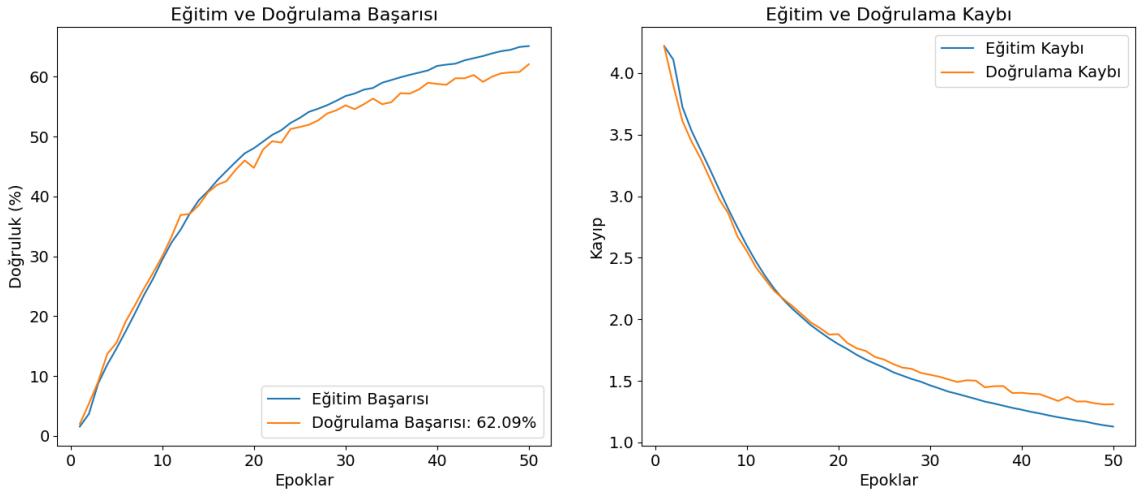
$$\text{Epok başına adım sayısı } = \frac{237089}{32} \approx 7409 \tag{15}$$



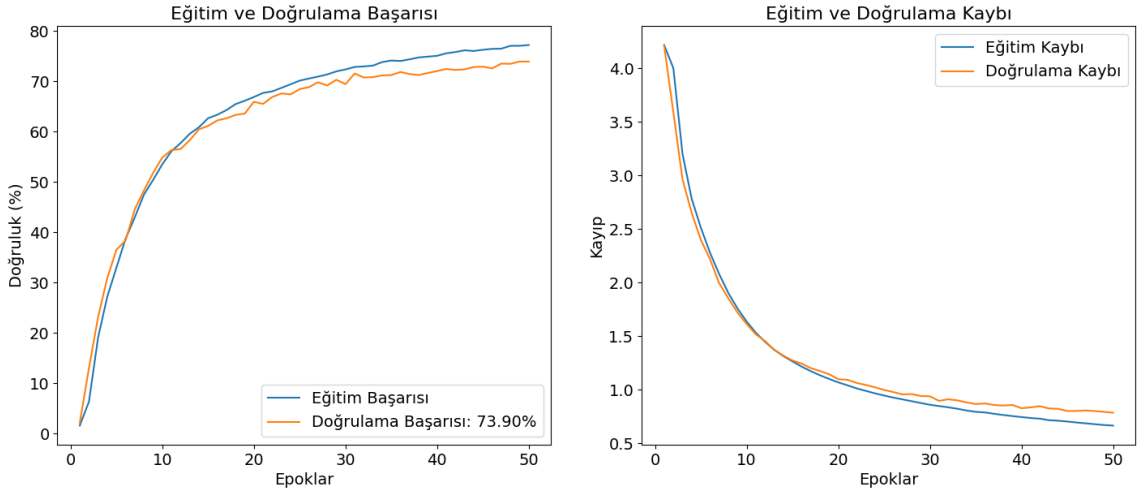
Şekil 17. Veri seti-3 karakter dağılım histogramı

4.7.5. Eğitim ve Doğrulama Başarısı ile Kaybı Grafikleri

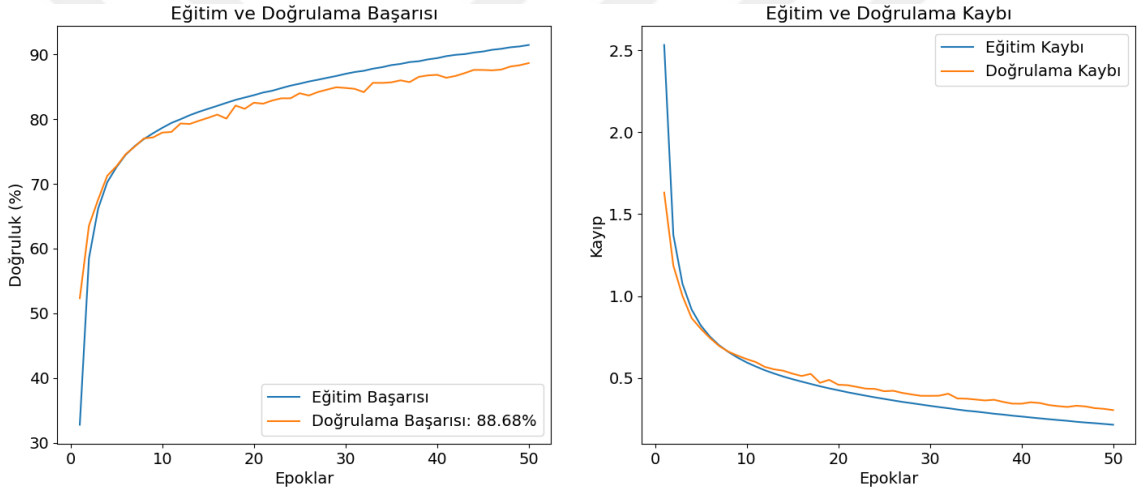
Bu bölümde eğitim ve doğrulama başarısı ile eğitim ve doğrulama kaybı grafiklerinin neyi ifade ettiği ve incelenen noktalar anlatılacaktır. Bu grafikler modelin performansını değerlendirmek için önemli bilgiler sunar. Bu verileri incelemek modelin ne kadar iyi çalıştığını ve potansiyel olarak hangi sorunlarla karşılaşabileceğini anlamaya yardımcı olur. Aşağıda Şekil 18 – 20'de veri seti-1, veri seti-2 ve veri seti-3 için eğitim ve doğrulama başarısı ile kaybı grafikleri verilmiştir.



Şekil 18. Veri seti-1 eğitim ve doğrulama başarısı ile kaybı grafikleri



Şekil 19. Veri seti-2 eğitim ve doğrulama başarısı ile kaybı grafikleri



Şekil 20. Veri seti-3 eğitim ve doğrulama başarısı ile kaybı grafikleri

Eğitim Başarısı: Eğitim başarısı genellikle doğruluk oranı ile ölçülür ve bir modelin eğitim veri seti üzerindeki doğruluğunu ifade eder. Doğruluk oranı modelin doğru tahmin yaptığı örneklerin toplam örneklere oranıdır.

Doğrulama Başarısı: Modelin doğrulama veri seti üzerindeki performansını gösterir. Eğitim sırasında modelin genelleşme yeteneğini anlamak için önemlidir.

Şekil 18 – 20'deki eğitim ve doğrulama başarı grafikleri incelendiğinde eğitim ve doğrulama başarılarının zamanla arttığı görülebilir. Modelin eğitim veri setleri üzerindeki doğruluğununun zamanla artması öğrenme sürecinin başarılı gittiği anlamına gelir. Eğitim verisi üzerinde iyi performans gösteren model doğrulama verisi üzerinde de benzer şekilde performans göstermiş olması modelin aşırı öğrenme yapmadığını ve yeni verilere iyi genellenebilir olduğunu gösterir.

Eğitim Kaybı: Eğitim kaybı modelin eğitim veri setindeki performansını hata miktarı üzerinden değerlendiren ölçümdür. Modelin yaptığı hataların büyüklüğünü ve modelin tahminlerinin ne kadar doğru olduğunu ifade eder. Bu değer kayıp fonksiyonu kullanarak hesaplanır ve modelin gerçek değerlerden sapmalarını ölçer.

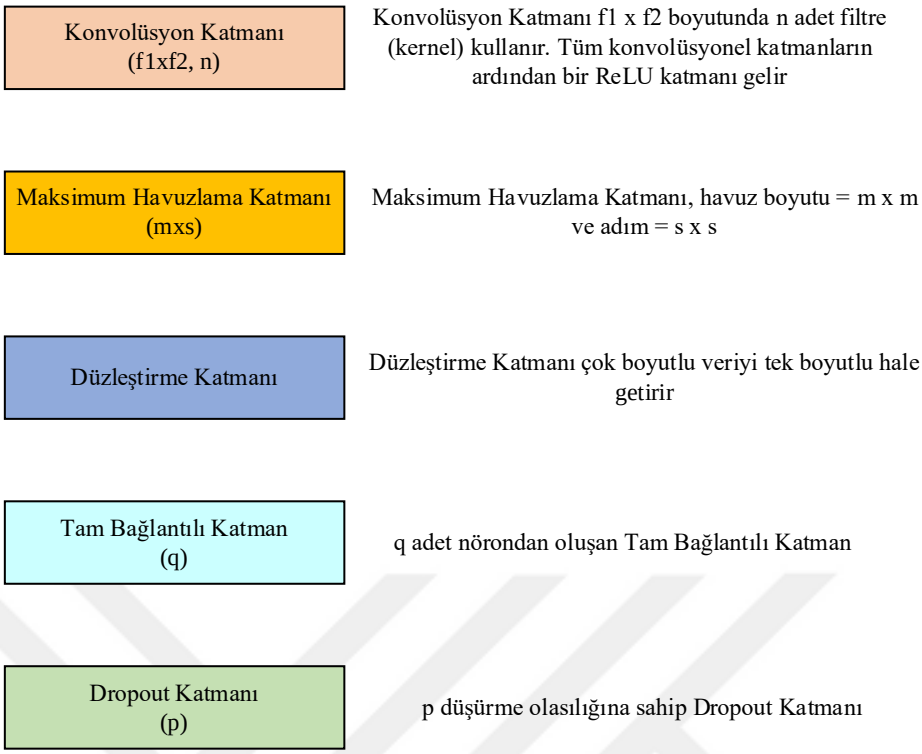
Doğrulama Kaybı: Modelin doğrulama veri seti üzerindeki performansını gösteren ölçümdür. Modelin doğrulama verileri üzerinde yaptığı hataların büyüklüğünü ifade eder.

Şekil 18 – 20'de eğitim ve doğrulama kaybı grafikleri incelendiğinde eğitim ve doğrulama kayıplarının birlikte düzenli bir şekilde azaldığı görülebilir. Modelin eğitim veri setleri üzerinde eğitim kaybının zamanla düşüyor olması modelin eğitim verisindeki örnekler üzerinde başarılı olduğunu gösterir. Eğitim kaybı azalırken doğrulama kaybının da azalması modelin veriyi öğrenmeye devam ettiğini ve iyi genelleme yaptığını gösterir.

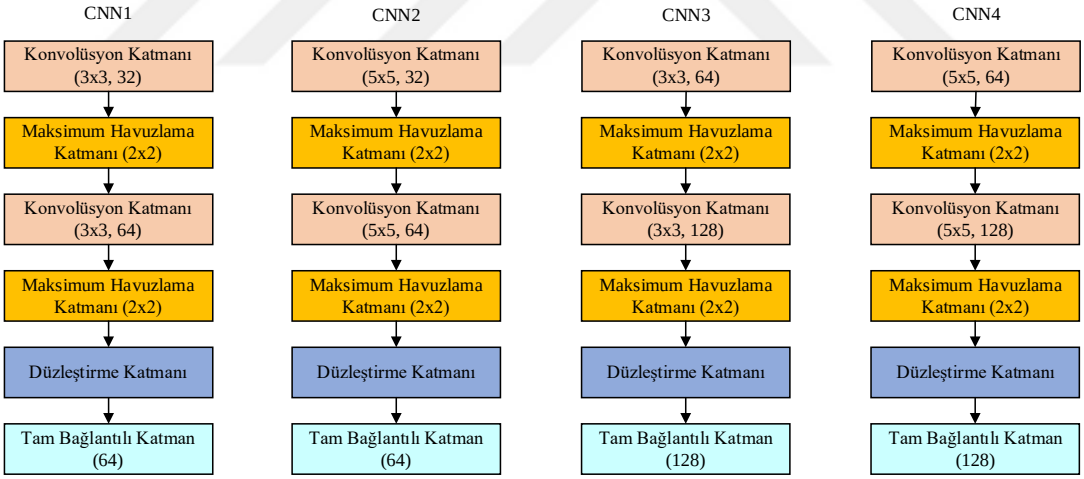
4.7.6. CNN Modelleri

Farklı katmanlar ve parametreler içeren farklı CNN modelleri veri seti-3 üzerinde karşılaştırılmak için kullanılmıştır. CNN'lerin topolojisini açıklamak için Şekil 21'de verilen semboller kullanılacaktır.

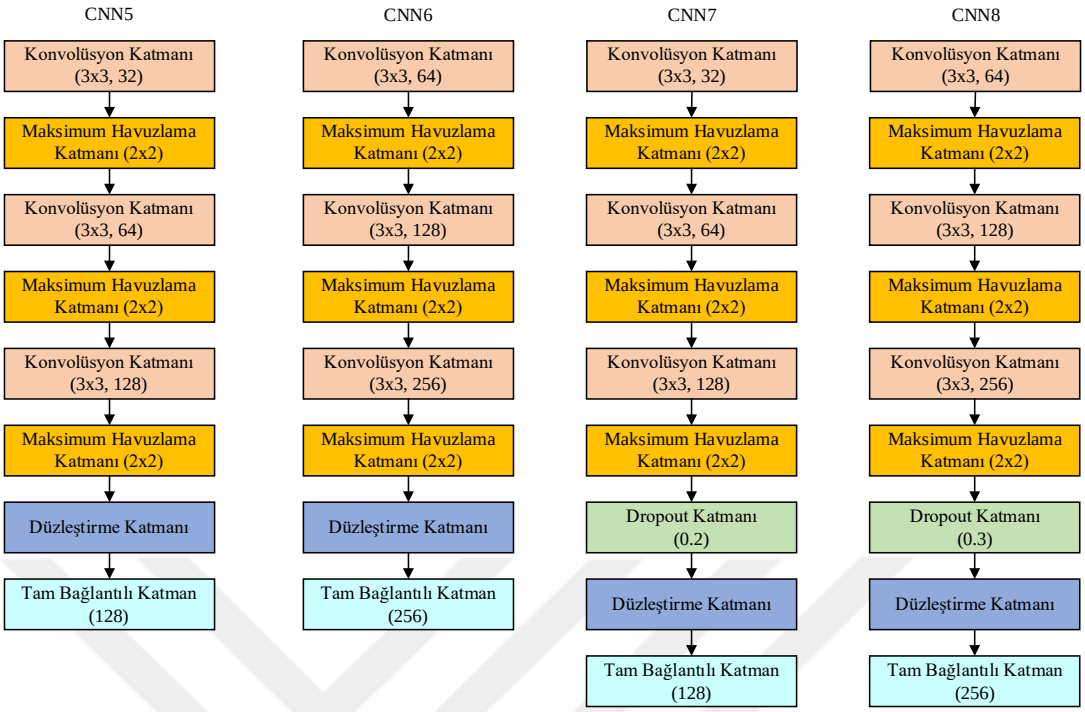
Oluşturulan CNN modelleri Şekil 22 – 25'te gösterilmiştir. Tüm konvolüsyonel katmanların ardından bir ReLU aktivasyon katmanı gelmekte ve tüm CNN modelleri bir sınıflandırma çıkış katmanı (Softmax) ile sona ermektedir. Bu katmanlar sürekli tekrar ettiğinden şekillerde gösterilmemiştir. Derin öğrenme modelleri oluşturmak, verileri işlemek ve görselleştirmek için TensorFlow ve Keras kütüphaneleri kullanılmıştır.



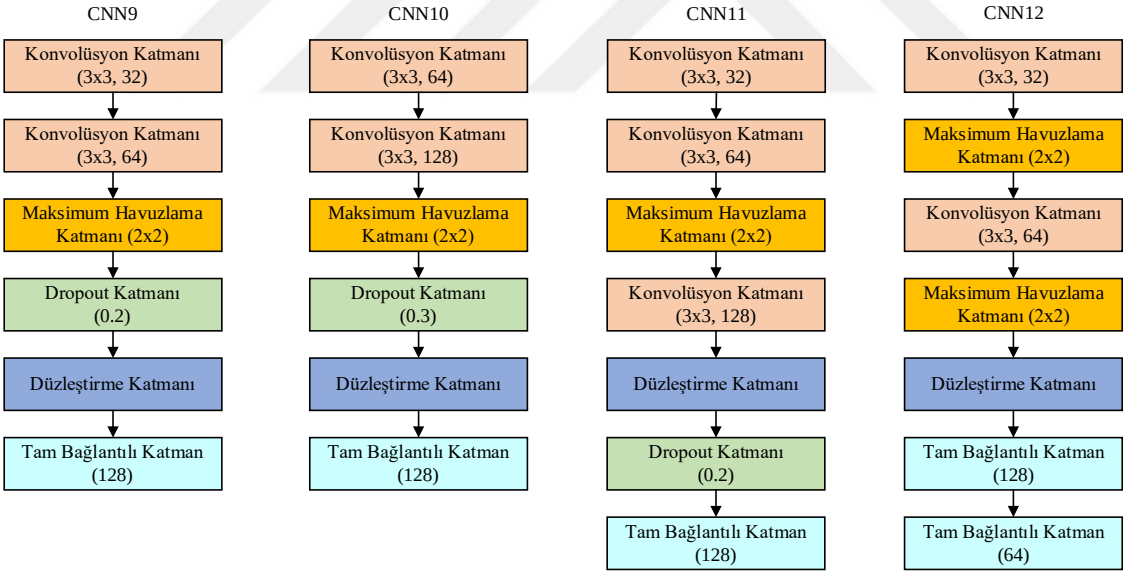
Şekil 21. CNN modellerini çizmek için kullanılan semboller



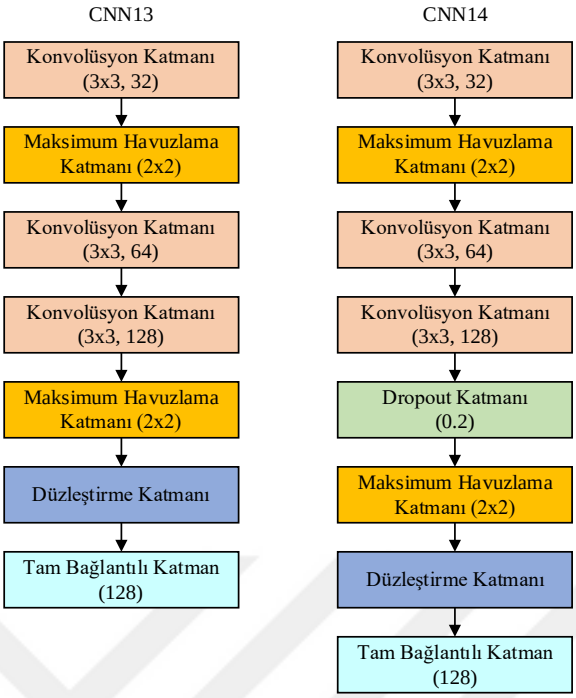
Şekil 22. CNN modelleri: CNN1-CNN4



Şekil 23. CNN modelleri: CNN5-CNN8



Şekil 24. CNN modelleri: CNN9-CNN12



Şekil 25. CNN modelleri: CNN13-CNN14

4.8. Test Aşaması

Bu bölümde CNN modellerinin veri setleri üzerindeki karakter tanıma performansını değerlendirmek için yapılan deneyler ele alınacaktır. Test verisi model tamamlandıktan sonra nihai performansını değerlendirmek için kullanılır. Modelin eğitim ve doğrulama süreçlerinde hiç kullanılmaz. Test verisi modelin gerçek dünya verileri üzerindeki başarısını tahmin etmek amacıyla kullanılır ve modelin yeni ve bilinmeyen verilerle nasıl bir performans sergileyeceğini gösterir. Başlangıçta veri setlerinde ön işleme ve veri artırma yöntemlerinin CNN modellerinin performansına olan etkisini incelemek amacıyla CNN5 modelinin veri seti-1, veri seti-2 ve veri seti-3 üzerindeki doğrulama ve test başarı sonuçları Tablo 1'de gösterilmiştir.

Tablo 1. CNN5 modelinin veri setleri üzerindeki doğrulama ve test başarısı

Veri Seti	**Doğrulama Başarısı %**	**Test Başarısı %**
Veri seti-1	62,09	62,37
Veri seti-2	73,90	73,56
Veri seti-3	88,68	88,61

Veri setinin son hali olan veri seti-3 üzerinde tüm CNN modelleri için test yapılmıştır. CNN modellerin doğrulama ve test başarısı Tablo 2'de karşılaştırıldığında her ikisinde en başarılı CNN11 modeli olduğu görülmüştür. Ardından CNN11 modelinin eğitim ve test performansını daha güvenilir bir şekilde değerlendirebilmek amacıyla Tablo 3'te verilen 5 katlı çapraz doğrulama uygulanmıştır. CNN11 modeli en fazla %93,86 test başarısı elde etmiştir ve 5 testin ortalaması %92,79 olmuştur.

Tablo 2. Tüm CNN'lerin doğrulama ve test başarısı

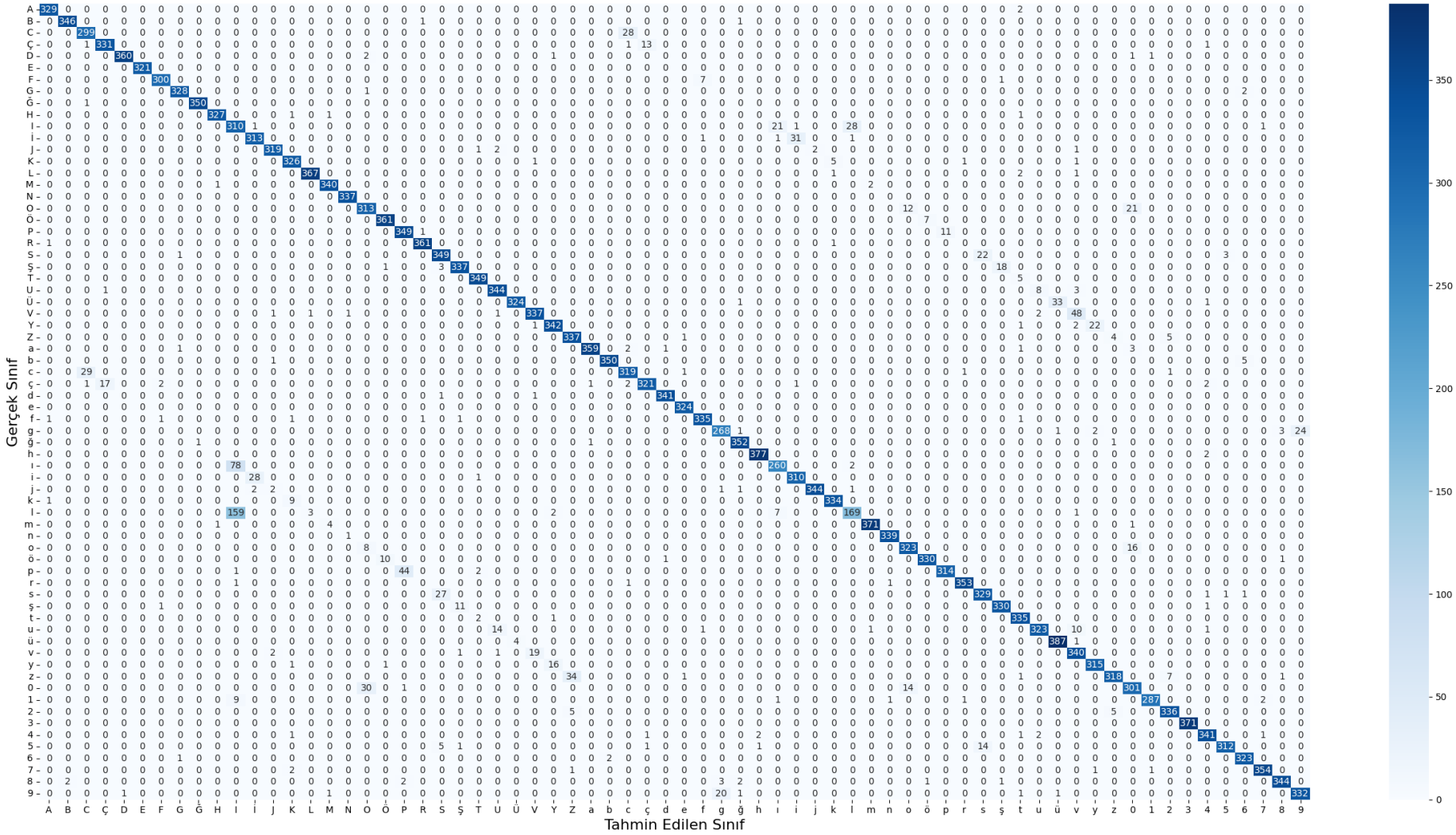
CNN modeli	**Doğrulama Başarısı %**	**Test Başarısı %**
CNN1	84,13	84,15
CNN2	87,02	87,14
CNN3	89,02	89,02
CNN4	92,03	92,18
CNN5	86,19	86,14
CNN6	92,00	92,37
CNN7	85,87	86,09
CNN8	91,40	91,30
CNN9	91,50	91,26
CNN10	93,14	93,02
CNN11	**93,95**	**93,86**
CNN12	87,56	87,31
CNN13	90,04	90,01
CNN14	93,29	93,44

Tablo 3. CNN11 modeli 5 katlı çapraz doğrulama ve test başarısı

CNN modeli	**Doğrulama Başarısı %**	**Test Başarısı %**
CNN11	91,91	91,55
	92,68	93,43
	93,32	93,32
	93,42	93,29
	92,61	92,36
Ortalama %	92,79	92,79

CNN modellerinin karşılaştırılması sonucu performans analizi yapmak için en iyi sonuçları veren CNN11 modeli seçilmiştir. CNN11 modelinin genel performansını değerlendirmek için aşağıda bazı performans metrikleri verilmiştir. Şekil 26'da verilen karmaşıklık matrisi sınıflandırma problemlerinde modelin performansını değerlendirmek için kullanılan bir araçtır. Özellikle çok sınıflı sınıflandırma görevlerinde modelin her bir sınıf için yaptığı tahminlerin doğruluğunu ve yanlışlıklarını görselleştirir ve modelin genel performansını değerlendirmeye yardımcı olur. Karmaşıklık matrisi aşağıdaki gibi bir yapıya sahiptir:

- Satırlar: Gerçek sınıfları temsil eder.
- Sütunlar: Modelin tahmin ettiği sınıfları temsil eder.
- Hücreler: Her hücre belirli bir gerçek sınıfa karşılık gelen tahmin edilen sınıfın sayısını gösterir.



Şekil 26. Karmaşıklık Matrisi

Kesinlik, duyarlılık ve F1 skoru metrikleri karmaşıklık matrisinden türetilir. Bu metriklerin her biri modelin sınıflandırma başarısını farklı bir açıdan değerlendirir ve modelin performansını anlamamıza yardımcı olur. Aşağıda bu metrikler açıklanmıştır:

Kesinlik: Doğru pozitif tahminlerin toplam pozitif tahminlere oranıdır. Modelin pozitif olarak tahmin ettiği her örneğin ne kadarının gerçekten pozitif olduğunu gösterir. Denklem 16'te verilmiştir.

Duyarlılık: Doğru pozitif tahminlerin gerçek pozitif örneklere oranıdır. Modelin gerçekten pozitif olan örnekleri ne kadar iyi bulduğunu gösterir. Denklem 17'da verilmiştir.

F1 Skoru: Kesinlik ve duyarlılığın harmonik ortalamasıdır. Denge sağlamak için kullanılır. Özellikle dengesiz sınıflar için önemlidir. Denklem 18'de verilmiştir.

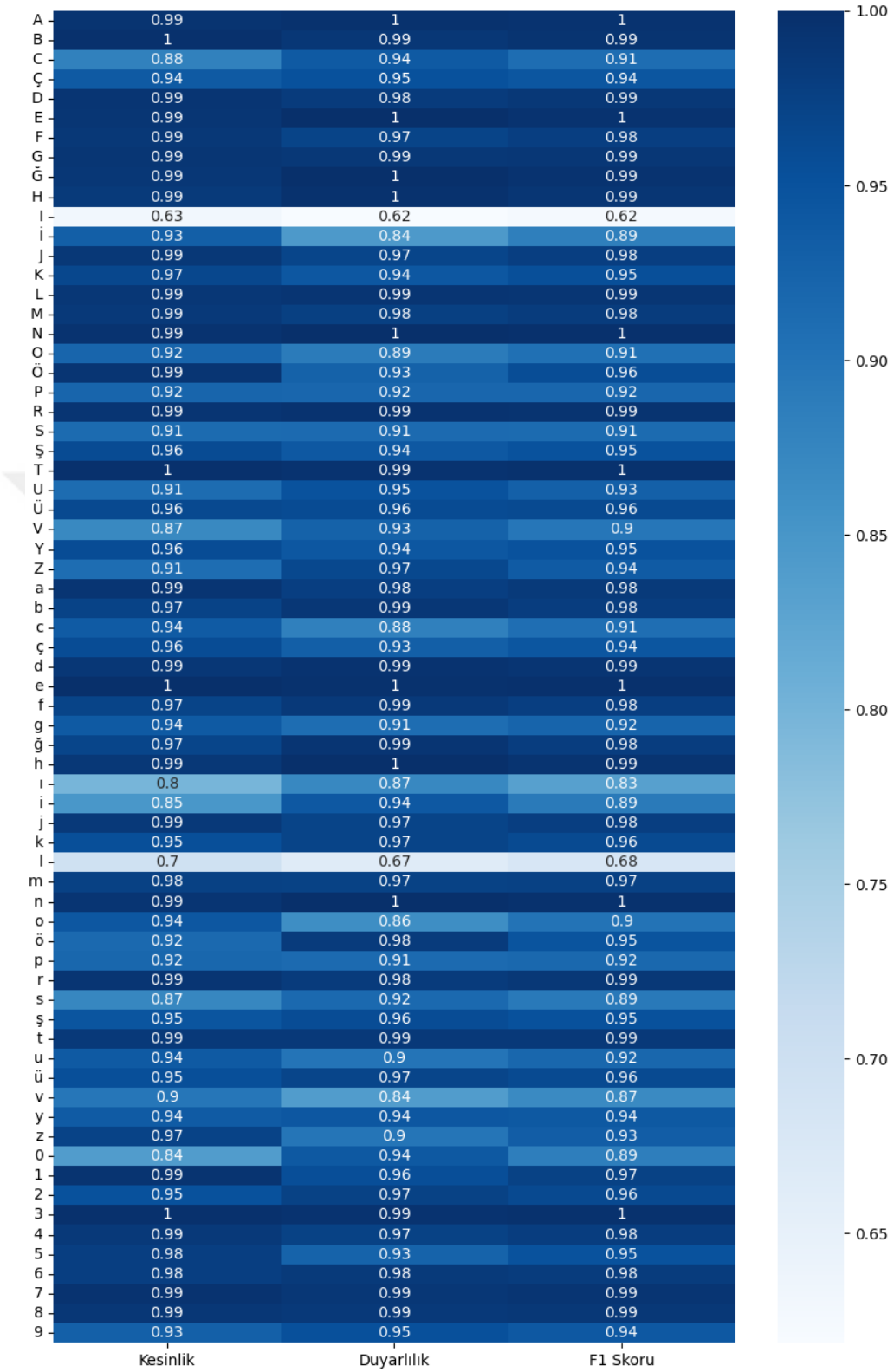
$$Kesinlik = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{16}$$

$$Duyarlılık = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{17}$$

$$F1\ Skoru = 2 \times \frac{Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \tag{18}$$

- TP (Gerçek Pozitif): Doğru bir şekilde pozitif olarak tahmin edilen örneklerdir.
- FP (Yanlış Pozitif): Yanlış bir şekilde pozitif olarak tahmin edilen örneklerdir.
- FN (Yanlış Negatif): Yanlış bir şekilde negatif olarak tahmin edilen örneklerdir.

Aşağıda Şekil 27'de veri setindeki her karakter için kesinlik, duyarlılık ve F1 skoru metrikleri verilmiştir. F1 skorları modelin o karakteri doğru bir şekilde tanıma yeteneğini gösterir. Yüksek F1 skorları modelin karakteri tanımada kesinlik ve duyarlılık açısından iyi performans gösterdiğini belirtirken, düşük F1 skorları ise modelin karakter ile ilgili tahminlerinde sorunlar olduğunu gösterir.



Şekil 27. Her karakterin kesinlik, duyarlılık ve F1 skorları

5. SONUÇ VE TARTIŞMA

Bu tez çalışmasında belgelerin dijitalleştirilmesi, arşivlenmesi ve belge işleme gibi birçok alanda sıklıkla kullanılan el yazısı karakter tanıma teknolojisinde Türk alfabesindeki özel harfleri içeren veri setlerinin kısıtlı olmasından dolayı literatürdeki bu boşluğu doldurmak amacıyla veri setleri oluşturulmuş ve el yazısı karakterleri tanıyabilen CNN tabanlı bir model önerilmiştir. Önerilen CNN tabanlı modelin performansını detaylı bir şekilde analiz edilmiştir.

Veri setinin oluşturulma aşamasında belgeler taratılırken düzgün yerleştirilmemesi ve donanımsal sorunlar gibi nedenlerden dolayı taratılan görüntülerin bir çoğu istenmeyen şekilde eğik olarak dijital ortama aktarılmıştır. Görüntülerin eğik bir şekilde taratılmış olması karakterlerin elde edilmesini zorlaştırmıştır. Bu nedenle karakterlerin kolayca elde edilebilmesi amacıyla ilk aşamada görüntüler düzleştirilmiş ve hizalanmıştır. Ardından karakterleri içeren kare şekiller tespit edilip görüntülerden kırpılarak karakterin ismiyle oluşturulan klasörlere kaydedilmiştir. Belgeler hazırlanırken bazı örneklerde karakterler kare şeklin sınırları dışına taşacak şekilde yazılması toplanan örnek sayısını azaltmıştır. Bu örneklerde kare tespiti başarılı olmamış ve karakterler çizgi üzerine yazıldığından şekil bozukluğuna uğramıştır. Bu örnekler düzgün bir şekilde elde edilememiştir.

Veri setini oluşturduktan sonra karakterler kontrol edilirken çok sayıda yazım yanlışı ile karşılaşılmıştır. Bazı belgelerde büyük harf yerine küçük harf yazılmış, veya tam tersi olmuştur. Bu gibi durumlar CNN modelinin performansını kötü etkileyebileceğinden tüm karakter görüntüleri kontrol edilmiş ve yazım yanlışları veri setinden çıkarılmıştır.

Bir diğer zorluk ise büyük ve küçük halleri şekil olarak aynı olan harflerin (C ve c, Ç ve ç, O ve o, Ö ve ö, S ve s, Ş ve ş, U ve u, Ü ve ü, Z ve z) belgelerde yakın boyutlarda yazılmasıdır. Birçok örnekte bir belgedeki büyük harfin başka bir belgedeki aynı harfin küçük halinden daha küçük ya da tam tersi şekilde yazıldığı tespit edilmiştir. HCR sistemlerinde HTR sistemlerinden farklı olarak bu harfler model tarafından sadece boyut farkıyla ayırt edilebilirler. HTR sistemlerinde ise bağlamsal bilgi, yazı stili, kelime yapısı ve anlam dikkate alınabilir. Aynı zamanda yazımları benzer şekilde olan karakterlerde

vardır (1 ile I, Z ile 2, 0 ile O gibi). Çeşitli metrikler kullanılarak CNN modellerinin performansı incelendiğinde en çok hata bu karakterlerde görülmüştür.

Sonuçlar ön işleme ve veri artırma yöntemlerinin CNN modellerinin el yazısı karakter tanıma performansını büyük ölçüde iyileştirildiğini göstermiştir. CNN modelleri veri setinin son hali üzerinde karşılaştırıldığında en iyi sonuçların peş peşe gelen konvolüsyon katmanlarında elde edildiği gözlemlenmiştir. Bu konvolüsyonel katmanlardan sonra maksimum havuzlama katmanı yer almamaktadır. Dropout katmanı derin ve karmaşık modellerde performansı artırırken daha basit modellerde düşüşe neden olmuştur.

İlerki çalışmalarda sistemin performansını iyileştirmek için veri seti çeşitliliği ve kalitesi artırılabilir, farklı ön işleme yöntemleri uygulanabilir. Ayrıca CNN model mimarisinin tasarımı ve farklı eğitim parametreleri ile daha iyi sonuçlar elde edilebilir.

6. KAYNAKLAR

Oak, P., Pandharkame, S., Munghate, S., & Kolge, V. (2021). Handwriting based personality analysis: An image processing based tool. *SSRN Electronic Journal*.

Shah, M. & Jethava, Gordhan. (2013). A literature review on hand written character recognition. Indian Streams Research Journal. 3. 1-19.

Kimura, F., Takashina, K., Tsuruoka, S., & Miyake, Y. (1987). Modified quadratic discriminant functions and the application to Chinese character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(1), 149–153.

Kang, L., Riba, P., Villegas, M., Fornés, A., & Rusiñol, M. (2021). Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. *Pattern Recognition*, *112*, 107790.

Grimsdale, R.L., Sumner, F.H., Tunis, C.J., & Kilburn, T. (1959). A system for the automatic recognition of patterns.

Eden, M. (1968). Handwriting Generation and Recognition, Chapter 5 in Kolers, P., and Eden, M. (Eds.), Recognizing Patterns, Cambridge, Mass.: MIT Press.

Suen, C.Y., Nadal, C., Mai, T.A., Legault, R. and Lam, L. (1990) Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. In: International Workshop Frontiers in Handwriting Recognition, Montreal.

Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 63–84.

Graves, A., & Schmidhuber, J. (2008). Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. *Neural Information Processing Systems*.

Liu, C.-L, Nakashima, K., Sako, H., & Fujisawa, H. (2003). Handwritten digit recognition: Benchmarking of state-of-the-art techniques. *Pattern Recognition*, *36*(10), 2271–2285.

Vidhale, B., Khekare, G., Dhule, C., Chandankhede, P., Titarmare, A., & Tayade, M. (2021). Multilingual text & handwritten digit recognition and conversion of regional languages into universal language using Neural Networks. *2021 6th International Conference for Convergence in Technology (I2CT)*.

AlKendi, W., Gechter, F., Heyberger, L., & Guyeux, C. (2024). Advancements and challenges in handwritten text recognition: A comprehensive survey. *Journal of Imaging*, *10*(1), 18.

Agrawal, V., Jagtap, J., & Kantipudi, M. V. V. P. (2024). Exploration of advancements in handwritten document recognition techniques. *Intelligent Systems with Applications*, *22*, 200358.

LeCun, Y., Cortes, C. and Burges, C.J.C. (1998) The MNIST Database of Handwritten Digits. New York, USA.

Bhatt, Preeti & Patel, Isha. (2018). Optical Character Recognition using Deep learning – A Technical Review. National Journal of System and Information Technology (NJSIT). 11.

Biswas, B., Bhadra, S., Sanyal, M. K., & Das, S. (2018). Online Handwritten Bangla Character Recognition Using CNN: A Deep Learning Approach (Vol. 695).

Rajyagor, B., & Rakholia, Dr. R. (2020). Handwritten character recognition using Deep Learning. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(6), 5815–5819.

Liu, C.-L., Yin, F., Wang, D.-H., & Wang, Q.-F. (2011). CASIA online and offline Chinese handwriting databases. *2011 International Conference on Document Analysis and Recognition*.

Shtaiwi, R. E., Abandah, G. A., & Sawalhah, S. A. (2022). End-to-end machine learning solution for recognizing handwritten Arabic documents. *2022 13th International Conference on Information and Communication Systems (ICICS)*, 180–185.

Luo, C., Zhu, Y., Jin, L., & Wang, Y. (2020). Learn to augment: Joint Data Augmentation and network optimization for text recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eltay, M., Zidouri, A., Ahmad, I., & Elarian, Y. (2021). Improving handwritten Arabic text recognition using an adaptive data-augmentation algorithm. *Lecture Notes in Computer Science*, 322–335.

Gao, H., Ergu, D., Cai, Y., Liu, F., & Ma, B. (2022). A robust cross-ethnic digital handwriting recognition method based on Deep Learning. *Procedia Computer Science*, *199*, 749–756.

Hayashi, T., Gyohten, K., Ohki, H., & Takami, T. (2018). A study of data augmentation for handwritten character recognition using Deep Learning. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*.

Hidayat, A. A., Purwandari, K., Cenggoro, T. W., & Pardamean, B. (2021). A convolutional neural network-based ancient Sundanese character classifier with data augmentation. *Procedia Computer Science*, *179*, 195–201.

Due Trier, Ø., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-A survey. *Pattern Recognition*, *29*(4), 641–662.

T. Caesar, J. M. Gloger and E. Mandler, "Preprocessing and feature extraction for a handwriting recognition system," Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93), Tsukuba, Japan, 1993, pp. 408-411.

Liu, Cheng-Lin & Marukawa, K. (2004). Normalization ensemble for handwritten character recognition. Proceedings- International Workshop on Frontiers in Handwriting Recognition, IWFHR. 69- 74. 10.1109/IWFHR.2004.76.

Paul, Nilima & Tunga, Harinandan. (2016). An Improved Method for Document Image Binarization.

Kılıçlar, B., & Makinacı, M. (2019). Recognition of handwritten source code characters with Deep Neural Networks. *4th International Symposium on Innovative Approaches in Engineering and Natural Sciences Proceedings*.

Patel, M., & Thakkar, S. P. (2015a). Handwritten character recognition in English: A survey. *IJARCCE,* 345–350.

Nikitha, A., Geetha, J., & JayaLakshmi, D. S. (2020a). Handwritten text recognition using Deep Learning. *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*.

Vinjit, B. M., Bhojak, M. K., Kumar, S., & Chalak, G. (2020a). A review on handwritten character recognition methods and Techniques. *2020 International Conference on Communication and Signal Processing (ICCSP)*, *7*, 1224–1228.

Peng, D., Jin, L., Wu, Y., Wang, Z., & Cai, M. (2019). A fast and accurate fully convolutional network for end-to-end handwritten Chinese text segmentation and recognition. *2019 International Conference on Document Analysis and Recognition (ICDAR)*.

Chaki, J., & Dey, N. (2018). *A Beginner's Guide to Image Preprocessing Techniques*.

Neetu Rani. (2017). Image processing techniques: A Review. *Journal on Today's Ideas-Tomorrow's Technologies*, *5*(1), 40–49.

Alginahi, Y. (2010). Preprocessing techniques in character recognition. *Character Recognition*.

Balci, B., Saadati, D., & Shiferaw, D. (2017). Recognition using Deep Learning.

Bansal, K., & Kumar, R. (2013). K-algorithm: A modified technique for noise removal in handwritten documents. *International Journal of Information Sciences and Techniques*, *3*(3), 1–8.

Kannan, R. J., Prabhakar, R., & Suresh, R. M. (2008). Off-line cursive handwritten Tamil character recognition. *2008 International Conference on Security Technology*, *1993*, 159–164.

Tin, Hlaing Htake Khaung. (2011). Removal of Noise Reduction for Image Processing.

Sauvola, Jaakko & Seppänen, Tapio & Haapakoski, Sami & Pietikäinen, Matti. (1997). Adaptive Document Binarization. Pattern Recognition. 33. 147-152 vol.1. 10.1109/ICDAR.1997.619831.

M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana and S. Apoorva, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2018, pp. 2319-2323.

Retsinas, G., Sfikas, G., & Nikou, C. (2023). Keyword spotting simplified: A segmentation-free approach using character counting and CTC re-scoring. *Lecture Notes in Computer Science*, 446–464.

Narayanan, Mahendran. (2023). Handwritten image augmentation.

Hosseini, MP., Lu, S., Kamaraj, K., Slowikowski, A., Venkatesh, H.C. (2020). Deep Learning Architectures. In: Pedrycz, W., Chen, SM. (eds) Deep Learning: Concepts and Architectures. Studies in Computational Intelligence, vol 866. Springer, Cham.

Sharifani, Koosha and Amini, Mahyar, Machine Learning and Deep Learning: A Review of Methods and Applications (2023). World Information Technology and Engineering Journal, Volume 10, Issue 07, pp. 3897-3904, 2023.

Hao, Z. (2019). Deep Learning Review and discussion of its future development. *MATEC Web of Conferences*, *277*, 02035.

Du, X., Cai, Y., Wang, S., & Zhang, L. (2016). Overview of deep learning. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Sewak, M., Sahay, S. K., & Rathore, H. (2020). An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, *17*(1), 182–188.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of Convolutional Neural Networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019.

Elngar, Ahmed & Arafa, Mohamed & Fathy, Amar & Moustafa, Basma & Mahmoud, Omar & Shaban, Mohamed & Fawzy, Nehal. (2021). Image Classification Based On CNN: A Survey. 6. 18-50. 10.5281/zenodo.4897990.

Van Dyk, D. A., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics, 10*(1), 1–50.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for Deep Learning. *Journal of Big Data, 6*(1).

O'Gara, S. & McGuinness, K. (2019). Comparing data augmentation strategies for deep image classification. *IMVIP 2019: Irish Machine Vision & Image Processing*, Technological University Dublin, Dublin, Ireland, August 28-30.

Mukhopadhyay, P., & Chaudhuri, B. B. (2015). A survey of Hough Transform. *Pattern Recognition, 48*(3), 993–1010.

Kingma, D.P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR, abs/1412.6980*.

Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines, 19*(1–2), 305–307.

ÖZGEÇMİŞ

İlköğrenimini Mimar Sinan İlköğretim Okulunda tamamladı. Lise öğrenimini Affan Kitapçıoğlu Anadolu Lisesi'nde tamamladı. 2014 yılında Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği bölümüne başladı. 2020 yılında Karadeniz Teknik Üniversitesi Elektrik Elektronik Mühendisliği Bölümünden mezun oldu. Halen Karadeniz Teknik Üniversitesi Elektrik Elektronik Mühendisliği Bölümünde Elektronik Mühendisliği Alanında yüksek lisans yapmaktadır. 2024 Mart ayından itibaren Karadeniz Teknik Üniversitesi Elektronik-Elektronik Mühendisliği Bölümü Biyomedikal Anabilim Dalında Araştırma Görevlisi olarak çalışmaktadır.