



T.C.

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI

**DERİN EVRİŞİMSEL SİNİR AĞLARINDA KISITLI AĞIRLIK BAŞLATIMI
İÇİN SEZGİSEL OPTİMİZASYON TABANLI YENİ AĞ MİMARİLERİNİN
SINIFLANDIRMA BAŞARIMLARININ İNCELENMESİ**

DOKTORA TEZİ

UĞUR TALAŞ

TEZ DANIŞMANI

PROF. DR. UĞUR YÜZGEÇ

İKİNCİ TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ BURAKHAN ÇUBUKÇU

BİLECİK, 2026

10670431

T.C.

BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĐİ DOKTORA PROGRAMI

**DERİN EVRİŐİMSEL SİNİR AĐLARINDA KISITLI AĐIRLIK BAŐLATIMI
İÇİN SEZGİSEL OPTİMİZASYON TABANLI YENİ AĐ MİMARİLERİNİN
SINIFLANDIRMA BAŐARIMLARININ İNCELENMESİ**

DOKTORA TEZİ

UĐUR TALAŐ

TEZ DANIŐMANI

PROF. DR. UĐUR YÜZGEÇ

İKİNCİ TEZ DANIŐMANI

DR. ÖĐR. ÜYESİ BURAKHAN ÇUBUKÇU

BİLECİK, 2026

10670431

BEYAN

Derin Evrişimsel Sinir Ağlarında Kısıtlı Ağırlık Başlatımı İçin Sezgisel Optimizasyon Tabanlı Yeni Ağ Mimarilerinin Sınıflandırma Başarımlarının İncelenmesi başlıklı doktora tezinin hazırlık ve yazım aşamasında bilimsel araştırma ve etik kurallarına uyduğumu, [Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Faaliyetlerinde Üretken Yapay Zekâ Kullanımına Dair Etik Rehberine](#) uygun olarak tez hazırladığımı, başkalarının eserlerinden yararlandığım bölümlerde bilimsel etik kurallarına uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, çalışmamın herhangi bir kısmının başka bir tez/dönem projesi olarak sunulmadığını, aksinin tespit edilmesi durumunda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Uğur TALAŞ

.../.../2026

İmza:

ÖNSÖZ

Bu tez çalışmasının hazırlanması sürecinde bilgi, deneyim ve yönlendirmeleriyle her zaman desteklerini esirgemeyen değerli tez danışmanlarım Prof. Dr. Uğur YÜZGEÇ ve Dr. Öğr. Üyesi Burakhan ÇUBUKÇU'ya en içten teşekkürlerimi sunarım. Hocalarım, araştırmanın tüm aşamalarında bilimsel bakış açısıyla yol göstermiş ve önemli katkılar sağlamıştır.

Araştırma sürecinde öneri ve yapıcı eleştirileriyle çalışmama katkı sunan tez izleme jüri üyeleri Prof. Dr. Cihan KARAKUZU ve Dr. Öğr. Üyesi Emre GÜNGÖR'e teşekkür ederim.

Doktora eğitimini almam için teşvik ve desteklerini esirgemeyen Daire Başkanım Murat Fidan'a ve onun nezlinde tüm Bilgi İşlem Daire Başkanlığı çalışanlarına, özellikle Yazılım Teknolojileri Şube Müdürlüğü'nde birlikte görev yaptığım Öğr. Gör. Yusuf MUŞTU, Öğr. Gör. Musa TURKAN ve Harun GUTER'e, ayrıca birlikte çalıştığım ve özellikle ders aşamasında bana destek veren Öğr. Gör. Semih KARACA'ya teşekkürlerimi sunarım.

Uzun zamandır parçası olduğum Öğrenci Bilgi Sistemi Geliştirme Ekibi'ne ve Öğrenci İşleri Daire Başkanı Sezer Kuyucu'ya, doktora çalışmalarına yoğunlaştığım dönemlerde iş yükümü paylaşarak, bana alan tanıyan Müh. Yavuz Biçici ve Mustafa Başdemir'e, her zaman çalışmalarımı destekleyip, beni motive eden Öğr. Gör. Fatih Aydın'a teşekkür ederim.

Bu süreçte bana arkadaşlık eden başta manevi kardeşim Yasin ADIGÜZEL'e, Semih ÇAĞLI ve Semih AKKERMAN'a ve tüm dostlarıma teşekkür ederim.

Desteğini her zaman yanımda hissettiğim aileme, sabırları için minnettarım. Özellikle yoğun çalışma dönemlerinde anlayış gösteren annem Veliye TALAŞ'a, abim Mesut TALAŞ'a ve yengem Gülden TALAŞ'a, kayın babam Resul TETİK'e, kayın validem Durdu TETİK'e, kardeşim Ezgi TETİK'e ve yeğenlerim Melek TALAŞ'a ve İpek TALAŞ'a teşekkür ederim.

Doktora sürecim boyunca stresimi, yoğunluğumu ve yorgunluğumu en çok paylaşan sevgili kızım İkra TALAŞ'a, canım oğlum Bilal Beyazıd TALAŞ'a ve her zaman sabırla ve anlayışla yanımda olup bana her konuda yardımcı ve yol arkadaşı olan kıymetli eşim Fatma Nur TALAŞ'a sabırları, sevgileri ve bana verdikleri güç için minnettarlığımı sunarım.

Son olarak, bugün aramızda olmasa da manevi olarak her zaman desteğini hissettiğim, hayatımda büyük bir yer eden babam Bilal TALAŞ'a sonsuz teşekkürlerimi sunarım.

Uğur TALAŞ

2026

ÖZET

Bu tez kapsamında, derin öğrenme modellerinin eğitim aşamasında parametrelerin rastgele başlatılmasından kaynaklanan performans kayıpları ele alınmıştır. Bu olumsuz etkileri azaltmaya yönelik yeni derin öğrenme yaklaşımları geliştirilmiştir. Çalışmanın temel hedefi, derin öğrenme ağlarında sıkça karşılaşılan yavaş yakınsama, yerel minimuma sıkışma ve dengesiz başlatılan ağırlıklar gibi sorunların en aza indirilmesidir. Bu doğrultuda, “kısıtlı ağırlık üretici” adı verilen yeni bir katman geliştirilmiş, bu katman CNN mimarisinde özellik çıkarım katmanı ile tam bağlantılı katman arasında konumlandırılarak tam bağlantılı katmanın daha uygun bir başlangıç noktasından eğitime başlamasını sağlamıştır. Çalışmanın ilk aşamasında D-CNN, C-CNN, S-CNN, RS-CNN ve M-CNN olmak üzere beş farklı kısıtlı CNN modeli önerilmiş; bu modeller MNIST, Fashion-MNIST ve CIFAR-10 veri kümeleri eğitilip test edilmiştir. MNIST veri kümesinde %99,16 doğruluk ile C-CNN, MNIST Fashion veri kümesinde %92,11 doğruluk ile D-CNN, CIFAR-10 veri kümesinde %71,69 doğruluk ile D-CNN en yüksek doğruluk elde etmiştir. Bu sonuçlara göre fark vektörü kullanan D-CNN ve C-CNN modellerinin diğer yaklaşımlara kıyasla daha yüksek doğruluk elde ettiği görülmüştür. Bu modellerdeki vektör seçim süreci SCO algoritması ile optimize edilip oluşturulan yeni karma modele OptiConNet adı verilmiştir. Literatürdeki benzer çalışmalardan farklı olarak, OptiConNet modelinde SCO algoritması ise doğrudan CNN parametrelerini optimize etmek yerine kısıtlı ağırlık üreticinin vektör seçimi optimize edilmiştir. OptiConNet modeli Fashion-MNIST ve Brain Tumor MRI veri kümeleri üzerinde test edilmiş ve özellikle Brain Tumor MRI veri kümesi üzerinde %96,56 doğruluk değerine ulaşılması modelin gerçek hayat problemlerine uygulanabilirliğini göstermiştir. OptiConNet modelinin tek aday tabanlı yapısının yerine, sürü temelli bir optimizasyon yaklaşımının benimseyen ABC algoritması modele entegre edilerek HyBee-CNN adı verilen yeni bir karma model geliştirilmiştir. HyBee-CNN modeli Fashion-MNIST veri kümesinde test edilmiş ve %92,75 doğruluk elde edilmiştir. Sonuç olarak, önerilen modellerin literatürdeki çalışmalarla karşılaştırıldığında üstün başarımlar göstermesi, kısıtlı ağırlık üretici yaklaşımı ile sezgisel optimizasyon algoritmalarının bir arada kullanıldığı karma mimarinin özgünlüğünü ortaya koymaktadır. Bu yönüyle çalışma, derin öğrenme modellerinin eğitimine alternatif ve etkili bir başlangıç stratejisi sunarak literatüre katkı sağlamıştır.

Anahtar Kelimeler: Yapay Zekâ, Derin Öğrenme, Kısıtlı ağırlık Üretici, Sezgisel Optimizasyon

ABSTRACT

In this thesis, the performance degradation caused by random parameter initialization during the training phase of deep learning models is thoroughly investigated. The primary objective of the study is to minimize common challenges encountered in deep neural networks, such as slow convergence, entrapment in local minima, and instability arising from poorly initialized weights. To address these issues, a novel layer referred to as the “constrained weight generator” is introduced. This layer is positioned between the feature extraction layer and the fully connected layer within a CNN architecture, enabling the fully connected layer to commence training from a more suitable and stable initialization point. In the initial phase of the study, five distinct constrained CNN models namely D-CNN, C-CNN, S-CNN, RS-CNN, and M-CNN are proposed and evaluated on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. The results demonstrate that C-CNN achieves the highest accuracy of 99.16% on the MNIST dataset, while D-CNN attains 92.11% accuracy on Fashion-MNIST and 71.69% accuracy on CIFAR-10. These findings indicate that the D-CNN and C-CNN models, which utilize difference vectors, outperform the other approaches in terms of classification accuracy. The vector selection process of these models is optimized using the SCO algorithm, leading to the development of a new hybrid model named OptiConNet. Unlike existing studies in the literature, where optimization algorithms are typically employed to directly tune CNN parameters, the SCO algorithm in OptiConNet is utilized to optimize the vector selection of the constrained weight generator. OptiConNet achieves 96.56% accuracy on the Brain Tumor MRI dataset, demonstrating real-world applicability. Furthermore, to overcome the limitations of the single candidate based structure of OptiConNet, a swarm-based optimization strategy is adopted by integrating the Artificial Bee Colony (ABC) algorithm into the model. This integration results in a new hybrid architecture termed HyBee-CNN. The HyBee-CNN achieves 92.75% accuracy on Fashion-MNIST. In conclusion, the superior performance of the proposed models compared to existing approaches in the literature highlights the originality and effectiveness of the hybrid architecture that combines the constrained weight generator with heuristic optimization algorithms. In this respect, the study contributes to the literature by offering an alternative and effective initialization strategy for training deep learning models.

Keywords: Artificial Intelligence, Deep Learning, Constrained Weight Generator, Heuristic Optimization

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	i
ÖZET.....	ii
ABSTRACT.....	iii
İÇİNDEKİLER.....	iv
TABLULAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
ALGORİTMA LİSTESİ.....	x
KISALTMALAR VE SİMGELER LİSTESİ.....	xi
1. GİRİŞ.....	1
1.1. Literatür Taraması.....	5
2. METOT.....	11
2.1. Evrişimli Sinir Ağı (Convolutional Neural Network - CNN).....	14
2.1.1. Özellik çıkarma (Feature extraction).....	15
2.1.2. Evrişim katmanı (Convolution layer).....	15
2.1.3. Dolgulama (Padding).....	17
2.1.4. Adım kaydırma (Stride).....	18
2.1.5. Havuzlama (Pooling).....	19
2.1.6. Tam bağlı katman (Fully connected network – FCN).....	20
2.2. Kısıtlı Evrişimli Sinir Ağları (Constrained CNN).....	22
2.2.1. Kısıtlı fark tabanlı evrişimli sinir ağı (D-CNN).....	24
2.2.2. Örnek tabanlı kısıtlı evrişimli sinir ağı (C-CNN).....	26
2.2.3. Toplama tabanlı evrişimli sinir ağı (S-CNN).....	28
2.2.4. Rastgele toplam tabanlı evrişimli sinir ağı (RS-CNN).....	30
2.2.5. Kısıtlı karma evrişimli sinir ağı (M-CNN).....	32

2.3. Sezgisel Optimizasyon Algoritmaları.....	34
2.3.1. Tek aday optimizasyon algoritması (SCO).....	35
2.3.2. Yapay arı koloni algoritması (ABC)	39
2.4. OptiConNet: Tek Aday Sınıf Seçimi ile Optimize Edilmiş Karma Kısıtlı CNN Modeli	41
2.5. HyBee-CNN: Yapay Arı Kolonisi ile Optimizasyonu Gerçekleştirilen Kısıtlı CNN Modeli	46
2.6. Veri Setleri.....	48
2.6.1. MNIST veri kümesi	49
2.6.2. MNIST Fashion veri kümesi	51
2.6.3. CIFAR-10 veri kümesi.....	52
2.6.4. Brain Tumor MRI veri kümesi.....	53
2.7. Başarım Değerlendirme Metrikleri	55
3. DENEYSEL SONUÇLAR VE TARTIŞMA.....	58
3.1. Kısıtlı CNN Model Sonuçları.....	58
3.1.1. Kısıtlı CNN modelleri MNIST veri kümesi sonuçları	59
3.1.2. Kısıtlı CNN modelleri MNIST Fashion veri kümesi sonuçları.....	63
3.1.3. Kısıtlı CNN modelleri CIFAR-10 veri kümesi sonuçları	69
3.1.4. Kısıtlı CNN modelleri genel değerlendirme	76
3.2. OptiConNet Model Sonuçları.....	78
3.2.1. OptiConNet modeli MNIST Fashion veri kümesi sonuçları.....	78
3.2.2. OptiConNet modeli Brain Tumor MRI veri kümesi sonuçları	84
3.3. HyBee-CNN Modeli MNIST Fashion Veri Kümesi Sonuçları	89
4. SONUÇLAR	95
KAYNAKÇA	98

TABLolar LİSTESİ

	Sayfa
Tablo 1.1 Literatür taramasına dayalı karşılaştırmalı analiz.	5
Tablo 2.1 Kısıtlı CNN modelleri çalışma yöntemleri özet tablo.	24
Tablo 2.2 Veri kümelerinin temel özellikleri.	49
Tablo 3.1 Kısıtlı CNN modellerinin MNIST veri kümesi test sonuçları karşılaştırılması.	60
Tablo 3.2 Kısıtlı CNN modellerinin MNIST Fashion veri kümesi test sonuçları karşılaştırılması.	64
Tablo 3.3 D-CNN modelinin MNIST Fashion veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.	69
Tablo 3.4 Kısıtlı CNN modellerinin CIFAR-10 veri kümesi test sonuçlarının karşılaştırılması.	71
Tablo 3.5 D-CNN modelinin CIFAR-10 veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.	75
Tablo 3.6 Kısıtlı CNN çalışma sürelerinin saniye cinsinden karşılaştırılması.	77
Tablo 3.7 OptiConNet modeli MNIST Fashion veri kümesi test sonuçlarının karşılaştırılması.	79
Tablo 3.8 OptiConNet modelinin MNIST Fashion veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.	83
Tablo 3.9 OptiConNet modeli Brain Tumor MRI veri kümesi test sonuçlarının karşılaştırılması.	85
Tablo 3.10 OptiConNet modelinin Brain Tumor MRI veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.	88
Tablo 3.11 Önerilen modellerin çalışma sürelerinin saniye cinsinden karşılaştırılması	93

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1 Çalışmanın metodolojik çerçevesi-1.....	12
Şekil 2.2. CNN Mimarisi ve katmanları	14
Şekil 2.3. Evrişim işlemi.	16
Şekil 2.4 Sıfır dolgu ve kenar tekrar dolgu örnekleri.....	18
Şekil 2.5. Havuzlama yöntemlerine ait uygulama örnekleri.	19
Şekil 2.6 Tam bağlı katman örneği.	20
Şekil 2.7 Kısıtlı CNN işlem akış öbek şeması.	22
Şekil 2.8 D-CNN ağındaki tam bağlı katman ağırlıkları ve eşik parametreleri belirleme işlemleri akış şeması.....	25
Şekil 2.9 C-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması. .	27
Şekil 2.10 S-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması.	29
Şekil 2.11 RS-CNN akış şeması.	31
Şekil 2.12 M-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması.	33
Şekil 2.13 SCO güncelleme fonksiyonunda kullanılan $w(t)$ 'nin değişim grafiği.....	36
Şekil 2.14 SCO akış şeması.	38
Şekil 2.15 OptiConNet modelinin mimarisi.	42
Şekil 2.16 OptiConNet mimarisinde ağırlık üretme sürecinin işlem adımları.....	44
Şekil 2.17 HyBee-CNN modelinin mimarisi.	47
Şekil 2.18 MNIST veri kümesi örnek görselleri.	50
Şekil 2.19 MNIST veri kümesi eğitim ve test sınıf dağılım grafiği.....	50
Şekil 2.20 MNIST Fashion veri kümesi örnek görselleri.	51
Şekil 2.21 MNIST Fashion veri kümesi eğitim ve test sınıf dağılım grafiği.	52

Şekil 2.22 CIFAR-10 veri kümesi örnek görselleri.	52
Şekil 2.23 CIFAR-10 veri kümesi eğitim ve test sınıf dağılım grafiği.	53
Şekil 2.24 Brain Tumor MRI veri kümesi örnek görselleri.	54
Şekil 2.25 MNIST veri kümesi eğitim ve test sınıf dağılım grafiği.	54
Şekil 2.26 Karmaşıklık matrisi örneği.	56
Şekil 2.27 AUC-ROC grafik örneği.	57
Şekil 3.1 C-CNN modelinin MNIST veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.	62
Şekil 3.2 C-CNN modeli MNIST veri kümesi test sonuçları AUC-ROC grafiği.	63
Şekil 3.3 D-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.	66
Şekil 3.4 D-CNN modeli MNIST Fashion veri kümesi test sonuçları AUC-ROC grafiği.	68
Şekil 3.5 D-CNN modelinin CIFAR-10 veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.	73
Şekil 3.6 D-CNN modeli CIFAR-10 veri kümesi test sonuçları AUC-ROC grafiği.	74
Şekil 3.7 Kısıtlı CNN modellerinin farklı veri kümelerinde doğruluk metriği için çubuk grafiği	76
Şekil 3.8 OptiConNet modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.	81
Şekil 3.9 OptiConNet modeli MNIST Fashion veri kümesi test sonuçları AUC-ROC grafiği.	82
Şekil 3.10 OptiConNet modelinin Brain Tumor MRI veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.	86
Şekil 3.11 OptiConNet modeli Brain Tumor MRI veri kümesi test sonuçları AUC-ROC grafiği.	87
Şekil 3.12. HyBee-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait sınıf bazlı kesinlik, duyarlılık ve F1 Skor dağılımı.	91
Şekil 3.13. HyBee-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait	

karmaşıklık matrisi..... 92

Şekil 3.14 HyBee-CNN modeli Brain Tumor MRI veri kümesi test sonuçları AUC-ROC grafiği..... 93

ALGORİTMA LİSTESİ

	Sayfa
Algoritma 2.1 D-CNN modelinin sözde kodu	26
Algoritma 2.2 D-CNN modelinin sözde kodu	28
Algoritma 2.3 S-CNN modelinin sözde kodu	30
Algoritma 2.4 RS-CNN modelinin sözde kodu	32
Algoritma 2.5 M-CNN modelinin sözde kodu.....	34
Algoritma 2.6 SCO algoritmasının sözde kodu	39
Algoritma 2.7 ABC algoritmasının sözde kodu.....	41

KISALTMALAR VE SİMGELER LİSTESİ

ABC: Artificial Bee Colony

AI: Artificial Intelligence

ANN: Artificial Neural Network

AUC-ROC: Area Under the Receiver Operating Characteristic

C-CNN: Sample Constrained CNN

CNN: Convolutional Neural Network

CELM: Constrained Extreme Learning Machine

CIFAR-10: Canadian Institute for Advanced Research 10

D-CNN: Constrained Difference CNN

DL: Deep Learning

ELM: Extreme Learning Machine

FCN: Fully Convolutional Network

HyBee-CNN: Yapay Arı Kolonisi ile Optimizasyonu Gerçekleştirilen Kısıtlı CNN Modeli

M-CNN: Constrained Mixed CNN

MRI: Magnetic Rezoanans Image

ML: Machine Learning

MNIST: Modified National Institute of Standards and Technology

MLP: Multi Layer Perceptron

OptiConNet: Tek Aday Sınıf Seçimi ile Optimize Edilmiş Karma Kısıtlı CNN Modeli

PSO: Particle swarm optimization

RS-CNN: Random Sum CNN

S-CNN: Constrained Sum CNN

SCO: Single Candidate Optimizer

1. GİRİŞ

Yapay zekâ (Artificial Intelligence – AI) kavramı, 1950’li yıllarda Alan Turing’in “Makineler düşünebilir mi?” sorusu ile temellenmiş ve günümüzde bilgisayar disiplini haricinde birçok disiplinde kullanılan, hatta kesişim noktası olan bir bilim alanı hâline gelmiştir (Turing 2007). Turing’in önerdiği AI sürecini başlatan bu soruyu sorduktan sonra, bu sorunun nasıl cevaplanabileceğiyle ilgili bir test geliştirmiştir. Turing testi özetle bir makinenin insan zekâsına benzer davranış sergileyip sergileyemeyeceğini ölçmeyi amaçlamıştır. Bu çalışmadan sonra bilgisayar sadece matematiksel hesaplar yapan bir makine değil, “insan düşüncesini taklit edebilecek bir makine olabilir mi?” sorusunun muhatabı olan bir cihaza dönüşmüştür. Bu potansiyel üzerine çalışmalar başlamıştır ve ilk olarak 1956 yılında Dartmouth Konferansı’nda John McCarthy tarafından “Artificial Intelligence” teriminin resmî olarak kullanılmasıyla birlikte, o tarihten günümüze kadar süre gelen yapay zekâ araştırmaları başlamıştır (McCarthy vd. 2006). Araştırmaların ilk yıllarında, günümüzde kullanılan modellerin çok uzağında, daha çok kurallara dayanan çalışmalar yapılırsa da, çok geçmeden 1958 yılında Frank Rosenblatt tarafından ilk algılayıcı (Perceptron) modeli geliştirilmiştir. Bu gelişme ile yapay zekâ yavaş yavaş günümüzdeki haline evrilmeye başlamıştır (Rosenblatt 1958).

1970’li yıllarda bilgi işleme yeteneklerinin artması ve istatistiksel yöntemlerin gelişmesiyle birlikte, “Makine Öğrenmesi” (Machine Learning – ML) kavramı ön plana çıkmıştır (Mitchell vd. 1986; Shinde ve Shah 2018). Makine öğrenmesi, sistemlerin geçmiş verilerden öğrenerek gelecekteki olayları tahmin etmesini sağlayan algoritmalara dayanmaktadır. Bu dönem yapay zekânın kurallara bağlı çalışmasından ziyade daha çok veriye dayalı bir bilim dalına dönüşmesinde önemli bir aşama olmuştur (Janiesch vd. 2021). 1980’lerde ise yapay sinir ağlarının (Artificial Neural Networks – ANN) yeniden gündeme gelmesiyle birlikte, bilgisayarların insan beyninin öğrenme yöntemini modelleme çabaları hız kazanmıştır (Rumelhart vd. 1986). Özellikle “Geriye Yayılım Algoritması” (Backpropagation) bu dönemde sinir ağlarının eğitilmesinde devrim yaratmıştır (Rumelhart vd. 1986). 2000’li yılların başında “Derin Öğrenme” (Deep Learning -DL) kavramı ortaya çıkmıştır (LeCun vd. 2015). Derin öğrenme, çok katmanlı sinir ağları kullanarak verilerden hiyerarşik özellik çıkarımı yapmayı hedefler (Jogin vd. 2018). Bu yapılar ile yapay zekâ, insan müdahalesine ihtiyaç duymadan karmaşık örüntüleri öğrenebilir hale gelmiştir. Özellikle görüntü, ses ve metin gibi yüksek boyutlu verilerin işlenmesinde derin öğrenme modelleri, geleneksel makine öğrenmesi yöntemlerine göre çok daha yüksek doğruluk değerlerine ulaşmıştır (Mavaie vd.

2023).

Makine öğrenmesi ve derin öğrenme alanındaki gelişmeler ile birlikte hızlı öğrenme yeteneğiyle öne çıkan Aşırı Öğrenme Makinesi (Extreme Learning Machine-ELM) tabanlı çalışmalar da artmıştır (Huang vd. 2004). Temel ELM yapısına ek olarak, çekirdek tabanlı Kernel-ELM, çok katmanlı hiyerarşik ELM (Multi-layer ELM) gibi farklı mimari ve algoritmalar da geliştirilmiş; bu modeller ELM'nin hızlı öğrenme üstünlüğünü korurken daha karmaşık veri yapıları üzerinde iyileştirilmiş doğruluk ve genelleme başarımı sunmuştur (Katılmış ve Karakuzu 2021; Kaur vd. 2023; Lu vd. 2018). Bu kapsamda Zhu ve arkadaşları tarafından geliştirilen Kısıtlı Aşırı Öğrenme Makinesi (Constrained Extreme Learning Machine-CELM) modeli, geleneksel ELM yaklaşımına kıyasla daha iyi genelleme yeteneği sunmuştur. CELM'de bağlantı ağırlıkları tamamen rastgele başlatılmak yerine, sınıf örneklerinin fark veya toplam vektörlerinden seçilen kısıtlı alt kümelerle belirlenmektedir. Bu yöntem, ELM'nin karakteristik hızlı öğrenme üstünlüğünü korurken doğruluk oranlarını anlamlı şekilde artırmaktadır (Zhu vd. 2015). Bu kısıtlı ağırlık üretim fikri, özellikle sinir ağı temelli yapıların başlangıç parametrelerinin etkisini ortaya koyması bakımından önemlidir. Bu çalışma kapsamında CNN içerisindeki Çok Katmanlı Algılayıcı (Multi Layer Perceptron-MLP) katmanına uyarlanacak olan kısıtlı ağırlık üretici bileşenin geliştirilmesine ilham vermiştir.

Derin öğrenme mimarileri arasında öne çıkan yapılardan biri Evrişimsel Sinir Ağı modeli (Convolutional Neural Network – CNN)'dir. CNN'lerin geleneksel yapay sinir ağlarından en temel farkı, yerel alıcı alanlara dayalı evrişim işleçleriyle girdi verisindeki hiyerarşiyi otomatik olarak öğrenme yeteneğidir (Alzubaidi vd. 2021a). Bu yapı sayesinde özellikle görüntü sınıflandırma, nesne tespiti ve tıbbi görüntü analizi gibi alanlarda görüntünün işlenmesinde yüksek başarı elde etmiştir (Kshatri ve Singh 2023). Özellikle Krizhevsky ve arkadaşları tarafından geliştirilen AlexNet modeli, ImageNet yarışmasında geleneksel yöntemlere kıyasla yüksek bir başarı artışı sağlamış ve CNN'lerin derin öğrenme alanındaki önemini ortaya koymuştur (Krizhevsky vd. 2012). LeCun ve arkadaşları el yazısı rakamlarını tanıma (Modified National Institute of Standards and Technology-MNIST) probleminde CNN tabanlı modeller kullanarak büyük bir başarı elde etmiş ve bu çalışmaları, modern CNN modellerinin temelini oluşturmuştur. Özellikle LeNet-5 modeli, evrişimsel ve havuzlama katmanlarının yanı sıra öğrenilebilir ağırlıklarla çalışan tam bağlı katmanları bir araya getirerek, görsel verilerin sınıflandırmasında derin öğrenmenin temel ilkelerini ortaya koymuştur (LeCun 1998a). CNN'lerin temel gücü, görsellerden manuel özellik çıkarımı yapmadan, çok katmanlı filtreleme

işlemleriyle anlamlı vektörler ile temsil edebilmesinden gelir. Ancak, iç yapısında bulunun bazı katmanlarda rastgele başlatılan parametreler nedeniyle bazen yerel minimumlarda takılma, yavaş yakınsama veya aşırı öğrenme gibi sorunlarla karşılaşılabilir (Cao vd. 2018).

Bu problemlere çözüm getirmek için literatürde birçok iyileştirme yöntemi geliştirilmiştir (Narkhede vd. 2022). Hinton ve Salakhutdinov derin ağların daha etkin eğitilmesi için ön eğitim yaklaşımını önermiştir (Hinton ve Salakhutdinov 2012). Daha sonraki yıllarda Ioffe ve Szegedy tarafından önerilen toplu normalizasyon (Batch Normalization) yöntemi, ağıın öğrenme sürecini istikrarlı hâle getirmiştir (Ioffe ve Szegedy 2015). Bununla birlikte, özellikle çok katmanlı mimarilerde karşılaşılan gradyan yok olması (vanishing gradient), yavaş yakınsama ve optimizasyon sürecinin yerel minimumlarda takılması gibi zorluklar, klasik gradyan tabanlı optimizasyon yöntemlerinin etkinliğini sınırlandırmaktadır. Bu sınırlamalar, son yıllarda CNN mimarilerinin başarımını artırmak amacıyla gradyan tabanlı yaklaşımların ötesine geçilmesine ve alternatif optimizasyon stratejilerine yönelim olmasına neden olmuştur. Bu bağlamda doğadan, biyolojiden veya toplu zekadan esinlenerek geliştirilen sezgisel algoritmalar, karmaşık, çok modlu ve türevi alınamayan hedef fonksiyonlarında global optimuma yakın çözümler üretme potansiyeliyle dikkat çekmektedir. Bu tür algoritmalar, özellikle gradyan bilgisi kullanılmayan veya yeterli olmadığı senaryolarda, geleneksel optimizasyon yöntemlerine kıyasla daha sağlam ve esnek bir alternatif sunabilmektedir (Ahmed ve Darwish 2021; Kaveh ve Mesgari 2023).

Son yıllarda yapılan çalışmalarda, sezgisel algoritmaların CNN eğitimiyle birleştirilmesi üzerine önemli başarılar elde edilmiştir. (Hassen vd. 2022; Kaveh ve Mesgari 2023). Wang ve arkadaşları Parçacık Sürü Optimizasyonu (PSO) ile CNN'in öğrenme oranını dinamik olarak ayarlamış ve modelin başarımında önemli fayda sağlamıştır (Wang vd. 2019). Benzer şekilde Zhang ve arkadaşları Yapay Arı Kolonisi (Artificial Bee Colony – ABC) algoritmasını kullanarak oluşturdukları ABC-Net isimli modelleri ile sınıflandırma başarımında anlamlı bir iyileşme elde etmiştir (Zhang vd. 2022). Ayrıca, Li ve arkadaşları GA-CNN isimli Genetik Algoritma (GA) tabanlı bir CNN optimizasyonu önererek, asfaltlarda oluşan hasar tespitinde %99 oranında doğruluk seviyesine ulaşmıştır. Bu çalışmalar sezgisel optimizasyon algoritmalarının derin öğrenme modellerinde potansiyel olarak kullanılabileceğini göstermektedir (Li vd. 2022).

Bununla birlikte, mevcut literatürde sürü tabanlı sezgisel algoritmaların çoğu yüksek hesaplama maliyetleri nedeniyle büyük veri kümlerinde verimlilik problemi

oluşturabilmektedir (Yildiz 2012). Bu bağlamda sürü tabanlı olmayan optimizasyon algoritmaları, daha düşük hesaplama maliyetleriyle sonuç elde etme potansiyeli taşımaktadır. Özellikle Shami ve arkadaşları tarafından geliştirilen Tek Aday Optimizasyon (Single Candidate Optimizer, SCO) algoritması, yalnızca tek bir çözüm adayı üzerinden çalışan hafif bir optimizasyon yaklaşımı olarak sunulmuş olup, düşük hesaplama maliyetine rağmen yüksek başarımlar göstermiştir (Shami vd. 2024).

Literatürdeki veri kümesinden elde edilen kısıtlı vektörlere dayalı başlangıç ataması yapan algoritmalara sahip çalışmalara bakıldığında daha çok ana veriler üzerinde ya da özellik çıkarım katmanında bir hesaplama yaparak CNN modelinin özellik çıkarım katmanının önüne eklenen bir karma yapıların oluşturulduğu görülmektedir (Bayar ve Stamm 2018). ELM modelinde olduğu gibi bağlantı ağırlıkları için yapılan kısıtlı bir mimari (Constrained ELM modeller) yapılan literatür araştırmalarında görülmemiştir. Bu tez çalışması kapsamında öncelikle literatürde daha önce denenmemiş olan bir yaklaşımla CNN içerisine kısıtlı ağırlık üretici katmanı eklenerek özgün bir mimari geliştirilmiştir. Ayrıca literatürde kullanılan sezgisel algoritmalar genellikle ya seçilen yapısal parametrelerin optimizasyonuna ya da doğrudan bağlantı ağırlıklarının optimizasyonuna yöneliktir. Geliştirilen CNN'e bütünleşik çalışacak şekilde kısıtlı bir CNN modelinin vektör seçimini optimize etmek için sezgisel optimizasyon algoritmalarıyla kullanıldığı yeni karma modeller önerilmiştir.

Bu tez çalışması, derin öğrenme modellerinin eğitim sürecinde rastgele atanan parametrelerin yol açtığı olumsuzlukları gidermek amacıyla, model parametrelerinin daha etkin bir şekilde belirlenmesini sağlayan yenilikçi mimari öneriler sunmaktadır. Çalışmada, geleneksel CNN mimarisine kısıtlı ağırlık üretici adı verilen özel bir ara katman entegre edilerek, ağırlık öğrenme sürecine daha etkin başlangıç noktalarından başlaması sağlanmıştır. Tezin ilk aşamasında, bu yaklaşımı temel alan beş farklı kısıtlı CNN modeli (D-CNN, C-CNN, S-CNN, RS-CNN ve M-CNN) geliştirilmiştir. İkinci aşamada ise bu modellerde kullanılan ağırlık üretim süreçlerini yönlendiren vektör seçim yönteminin optimizasyonu amacıyla, bir sezgisel optimizasyon (SCO algoritması) mimariye dahil edilmiştir. Elde edilen tümleşik yapıya OptiConNet adı verilmiştir.

OptiConNet'in özgünlüğü, iki temel katkıya dayanır: Kısıtlı ağırlık üreticinin CNN mimarisi içindeki konumlandırılması ve tek aday optimizasyon algoritmasının doğrudan CNN parametrelerini değil, bu üretici bileşenin hesaplama dinamiklerini optimize etmesi. Bu yönleriyle önerilen yaklaşım, literatürde bugüne kadar rapor edilmemiş, özgün bir karma yapı

sunmaktadır. Çalışma derin öğrenme sistemlerinde sezgisel algoritmaların yalnızca doğrudan parametre optimizasyonu amacıyla değil, model içi yardımcı bileşenlerin kaynak kullanımını ve başlangıç koşullarını iyileştirme amacıyla entegre edilmesine yeni bir bakış açısı getirmiştir.

Önerilen OptiConNet modelinde kullanılan birey tabanlı sezgisel optimizasyon yaklaşımı (SCO) yerine sürü tabanlı bir optimizasyon algoritması olan Yapay Arı Kolonisi (Artificial Bee Colony, ABC) entegrasyonu gerçekleştirilmiş ve böylece yeni bir model olan HyBee-CNN geliştirilmiştir. Burada amaç sürü tabanlı (ABC) ve tek adaya dayanan (SCO) iki sezgisel algoritmanın aynı problemin çözümündeki başarımının test edilmesidir. HyBee-CNN'in genel çalışma akışı OptiConNet ile benzerlik göstermesine rağmen, kullanılan sezgisel algoritmaların çözüm arama stratejileri temelde farklılık arz etmektedir. Ayrıca ABC algoritmasının literatürde uzun süredir yaygın olarak kullanılan, olgunlaşmış ve güvenilir bir optimizasyon yöntemi olması, SCO tabanlı modele bir alternatif olmasına sebep olmuştur.

Bu bölümün devam eden kısmında literatürde yapılan çalışmalara yer verilmiştir. Tez kitabının ikinci bölümünde önerilen modellerin detayları verilirken, deneysel sonuçlar ve tartışma bölümünde çalışma kapsamında elde edilen sonuçlar verilmiştir. Sonuç bölümünde kapsamında elde edilen bulguların kapsamlı bir değerlendirme yapılmıştır.

1.1. Literatür Taraması

Literatürde incelenen çalışmalar yıl, kullanılan veri kümesi, uygulanan yöntem ve elde edilen kazanımlar açısından Tablo 1.1'de özetlenmiştir.

Tablo 1.1 Literatür taramasına dayalı karşılaştırmalı analiz.

Yıl / yazar	Çalışma / Yöntem	Veri Kümesi / Sonuç
(Pathak vd. 2015)	Zayıf Gözetimli Kısıtlı Evrişimli Sinir Ağları CCNN, çıktı uzayına lineer kısıtlar ekleyerek Resim Seviye İşaretlerle piksel seviyesinde bölümlenme öğrenebilen yenilikçi bir yöntemdir. Latent dağılım + CNN optimizasyonu sayesinde zayıf gözetimde iyi sonuçlar elde edilmiştir.	Doğruluk metriği; Pascal Voc12 : %54

Tablo 1.1 (Devamı) Literatür taramasına dayalı karşılaştırmalı analiz.

(Bayar ve Stamm 2018)	Kısıtlı Evrişimli Sinir Ağları: Genel Amaçlı Görüntü Manipülasyonu Tespitine Yönelik Yeni Bir Yaklaşım. Çalışmada kısıtlı evrişim katmanını içeren MISLnet modeli önerilmiştir.	Çalışmada oluşturulan veri kümesinde %99,36 doğruluk elde edilmiştir.
(Weng vd. 2018)	Kısıtlı Aşırı Öğrenme Makinesi (CELM) Kullanılarak CNN Tabanlı Arazi Kullanım Sahne Sınıflandırması Çalışmada CNN tarafından çıkarılan derin özellikler, CELM tarafından sınıflandırılarak hem çok hızlı hem de yüksek doğrulukta arazi kullanım sınıflandırması gerçekleştirilmiştir.	Doğruluk metriği AID: %99,70 (30 sınıf içermektedir), Sydney: %98,76 (7 sınıf içermektedir)
(Kervadec vd. 2019)	Zayıf Gözetimli Bölümlendirme için Kısıtlı-CNN Kayıpları CNN kayıp değerlerinin kısıtlanmasıyla oluşturulan ENet ve Residual U-Net yöntemleri kullanılmıştır.	Doğruluk metriği; ACDC LV MRI :%87 Vertebral Body MRI : %86 Prostate MRI :%83
(Yang vd. 2019)	Kısıtlı R-CNN: Genel Bir Görüntü Manipülasyonu Tespit Modeli Kısıtlı R-CNN modeli ile iki aşamalı şekilde hem bölümlendirme hem de sınıflama yapılmıştır. Kısıtlama işlemi evrişim katmanında kullanılmıştır.	F1 Skor metriği; Nist16: 0,927 Cover: 0939 Columbia 0,861 Casia 0,475
(Andonie ve Florea 2020)	CNN Hiperparametre Optimizasyonu için Ağırlıklı Rastgele Arama Yöntemi Çalışmada CNN hiperparametre optimizasyonunda kullanılan klasik rastgele arama yöntemine iyileştirme getiren WRS (Weighted Random Search) yöntemini önerilmektedir.	WRS yöntemi CIFAR-100 veri kümesinde %85 doğruluk elde etmiştir.
(Zhang vd. 2020)	ACCL: Zayıf Gözetimli Tıbbi Görüntü Bölümlendirme için Kısıtlı-CNN Kaybı ACCL (Adversarial Constrained-CNN Loss) zayıf gözetimli tıbbi görüntü bölümlendirme için kullanılan üretici SCCL yöntemine alternatif bir model önerilmiştir.	ACCL ortalama DSC: 75,4% SCCL ortalama DSC: 64,0% İyileşme = +11,4%

Tablo 1.1 (Devamı) Literatür taramasına dayalı karşılaştırmalı analiz.

(Cheng vd. 2021)	Nesne ve Sahne Bağlamı Kısıtlı CNN Tabanlı Uzaktan Algılamada Hedef Tespiti Çalışmada uzaktan algılama görüntülerinde hedef tespiti için hem nesne düzeyi bağlamı hem de sahne düzeyi bağlamı kullanan iki kanallı bir Bağlam Kısıtlı CNN modeli önermektedir.	Doğruluk metriği; NWPU : %91,7 HRRSD : %73,9
(Oh vd. 2021)	Bina Sınırlarını Dikkate Alan Dayanıklı Kapalı Alan Konumlandırma için Kısıtlı CNN Bu çalışmada, kablosuz cihaz parmak izi tabanlı kapalı alan konum tahmininde konumun bina dışına taşması sorununu gidermek için, bina sınırı bilgisi eklenmiş yeni bir CNN tabanlı model olan C-CNNLoc önerilmektedir.	C-CNNLoc modeli, ortalama konumlandırma hatasını 11,753 metreden 8,870 metreye düşürerek konumlandırma doğruluğu anlamlı şekilde artırmıştır.
(Dogaru ve Dogaru 2021)	NL-CNN: Doğrusal Olmayan Evrişime Dayalı Kaynak-Kısıtlı Derin Öğrenme Modeli Çalışmada düşük hesaplama maliyetine sahip cihazlarda çalışmak üzere tasarlanmış, doğrusal olmayan evrişimin taklit edildiği kompakt bir derin öğrenme modeli olan NL-CNN mimarisini önerilmiştir. Model, hızlı eğitim süresi, az parametre kullanımı ve yüksek sınıflandırma doğruluğu sağlamaktadır.	Doğruluk metriği; MNIST: %99,7, GTSRB: %98,97, FER2013: %63,86,
(Hussain vd. 2022)	Kablosuz Sensör Ağlarında Saldırı Tespiti için WOA-ABC Karma Modeli ve Önerilen CNN Tabanlı IDS Çalışmada karma WOA-ABC özellik seçimi ve 15 katmanlı özel CNN mimarisi kullanılarak WSN saldırıları yüksek doğruluk ve düşük yanlış alarm ile tespit edilmiştir.	NSL-KDD 99 veri kümesi için PSO-IDS ve klasik ML yöntemlerine kıyasla %76–87 arasında daha hızlı %3–20 aralığında daha yüksek doğruluk elde etmiştir.

Tablo 1.1 (Devamı) Literatür taramasına dayalı karşılaştırmalı analiz.

(Yalçın vd. 2022)	Elektrikli Araçlar için Lityum-İyon Bataryalarda Isı Üretim Oranı ve Gerilim Dağılımlarının Tahmini ve Optimizasyonu için CNN-ABC Modeli Çalışmada lityum-iyon bataryalarda ısı üretim oranı ve gerilim bir karma derin öğrenme modeli CNN-ABC önermektedir. Model, 2D-3D CNN katmanlarından elde edilen özellikleri ABC algoritmasıyla optimize ederek hata değerlerini önemli ölçüde azaltmaktadır.	CNN-ABC modeli, ısı üretim oranı ve gerilim tahminlerinde çok düşük hata oranlarına ulaşmıştır. Isı üretim Oranında: RMSE=1,38 Gerilimde: RMSE=3,55
(Kervadec vd. 2022)	Kısıtlı Derin Ağlar: Log-Barikat uzantıları(Log-Barrier extension) ile Lagrange Optimizasyonu Çalışmada derin sinir ağlarına eşitsizlik kısıtları uygularken klasik yöntemlerinin ve standart Lagrange yönteminin modern CNN'lerde başarısız olduğu için Log-Barikat uzantıları yaklaşımı önererek kısıtlı CNN optimizasyonu yapılmıştır.	Tıbbi görüntü işleme çalışmalarında kullanılan benzerlik ölçütü metriği DSC; Synthetic circles 0,945 DSC, Promise12: 0,813 DSC
(Belciug 2022)	DE kullanılarak öğrenme mimarilerinin oluşturulması. DE ile CNN parametreleri filtre sayısı havuzlama tipi tam bağlı katman boyutları gibi parametreleri optimize etmişlerdir.	Doğruluk metriği; Brain Cancer : %90,04 Lung and Colon Cancer Histopathological: %99,5 Maternal-fetal ultrasound %78,73
(Yan vd. 2023)	Lityum-İyon Bataryalarda Şarj Durumu (SOC) Tahmini İçin Bilgi-Kısıtlanmalı CNN-BiLSTM Modeli Model, fiziksel bilgi kısıtları içeren kayıp fonksiyonuyla lityum-iyon bataryalarda SOC tahmininde düşük veri koşullarında bile yüksek doğruluk ve daha düşük hata varyantı sağlamaktadır.	Panasonic veri kümesi: MSE %0,33-%0,88
(Rosero-Montalvo vd. 2024)	Donanım Kısıtlı IoT Cihazlarında Optimize Edilmiş CNN Mimarilerinin Karşılaştırılması Çalışma, CNN modellerini uç cihazlarda çalıştırmak için kapsamlı bir kıyaslama sunarak doğruluk boyut gecikme, güç ilişkisini detaylı biçimde analiz etmektedir.	EfficientNet-B0 doğruluk ve verimlilikte en dengeli model olmuştur. MobileNetV2 en hızlı ve en hızlı model olmuştur. VGG-16 ve ResNet-50 yavaş modellerdir.

Tablo 1.1 (Devamı) Literatür taramasına dayalı karşılaştırmalı analiz.

(Nagaveni vd. 2024)	Alzheimer Evrelerinin Sınıflandırılması için ABC Optimizasyonlu CNN Modeli	Doğruluk metriği ; CNN: %86,21
	Çalışma, Alzheimer hastalığının MRI görüntülerinden beş farklı evresini (EMCI, LMCI, MCI, AD, CN) sınıflandırmak için bir CNN modeli tasarlamakta ve modelin kritik hiperparametrelerini ABC algoritması ile optimize edilmiştir.	ABC-CNN : %94,00 F1 Skor metriği; CNN: %82,85 ABC-CNN : %91,00
(Çetiner ve Metlek 2025)	ABC+CNN-SH: ABC Optimizasyonu ve CNN Tabanlı Yeni Karma Bir Sınıflandırma Algoritması ile Peru Kahve Yaprak Hastalıklarının Tespiti	Doğruluk metriği; CoLeaf-DB : %95 AcLeaf-DB : %97 RoCoLe: %97 Soroti-DB : %92
	Çalışmada Peru kahve yapraklarındaki mineral eksikliği kaynaklı hastalıkları tespit etmek için ABC optimizasyonu ile hiperparametreleri optimize edilen hafif bir CNN mimarisi (ABC+CNN-SH) önermektedir.	
(Inik 2025)	SwarmCNN: PSO ve ABC Meta-sezgisel Algoritmaları Kullanarak CNN Hiperparametre Optimizasyonu için Etkili Bir Yöntem	Doğruluk metriği; MNIST: %99,58 CONVEX : %96,92
	Çalışmada CNN’lerde hem katman sayısı hem de katman parametreleri aynı anda optimize edebilen PSO (Parçaçık Sürü Optimizasyon Algoritması) ve ABC kullanılan yeni bir yöntem önermektedir.	MNIST Fashion : %93,47 MNIST BI : %96,39

Tablo 1.1’de sunulan literatür taramasında ağırlıklı olarak 2020-2025 yılları arasında gerçekleştirilen çalışmalar incelenmiştir. Bu kapsamda “Kısıtlı CNN”, “Hiperparametre Optimizasyonu”, “ABC–CNN Karma Yaklaşımları” ve “CNN ile Sezgisel Optimizasyon” anahtar kelimeleri üzerine odaklanılmıştır.

Kısıtlı CNN alanındaki çalışmalar incelendiğinde, çoğunlukla evrişim katmanları üzerinde gerçekleştirilen yapısal iyileştirmelerin ele alındığı görülmektedir. Bunun yanı sıra, veri kümesi üzerinde belirli kısıtlamalar uygulanarak gerçekleştirilen çalışmalar da literatürde yer almaktadır. Ancak, tez kapsamında önerilen Kısıtlı CNN modellerinde kullanılan “kısıtlı ağırlık üretici” yöntemi ile bağlantı ağırlıklarının hesaplanmasına yönelik yaklaşımın mevcut çalışmalarla doğrudan benzerlik göstermediği tespit edilmiştir. Literatürde, tez kapsamında

önerilen modellere en yakın çalışma, CNN'in tam bağı katmanındaki MLP bileşeninin çıkarılarak yerine Kısıtlı ELM yapısının entegre edildiği çalışmadır (Weng vd. 2018). Bu çalışma, bağlantı ağırlıklarına odaklanması bakımından benzerlik gösterse de, tezde önerilen modellerin yapısal olarak daha farklı ve özgün bir yaklaşım sunduğu anlaşılmaktadır.

Sezgisel optimizasyon algoritmalarına ilişkin yapılan incelemede SCO ile CNN'in karma olarak kullanıldığı bir çalışmaya rastlanmamıştır, ancak ABC algoritmasının CNN ve diğer derin öğrenme modelleriyle yaygın olarak kullanıldığı görülmüştür. Literatürde CNN parametrelerinin optimize edilmesine yönelik çalışmaların yanı sıra, tam bağı katmandaki bağlantı ağırlıklarını doğrudan optimize etmeyi hedefleyen araştırmalar da bulunmaktadır. Ancak yapılan değerlendirme sonucunda, tez kapsamında önerilen OptiConNet ve HyBee-CNN modellerinin literatürdeki yaygın yaklaşımlardan farklı olarak doğrudan CNN mimarisinin kendisini değil, model içerisindeki kısıtlı ağırlık üretici modülünü optimize ettiği görülmektedir. Bu yönüyle önerilen yöntemlerde sezgisel algoritmaların kullanımının literatüre özgün bir katkı sunduğu değerlendirilmektedir.

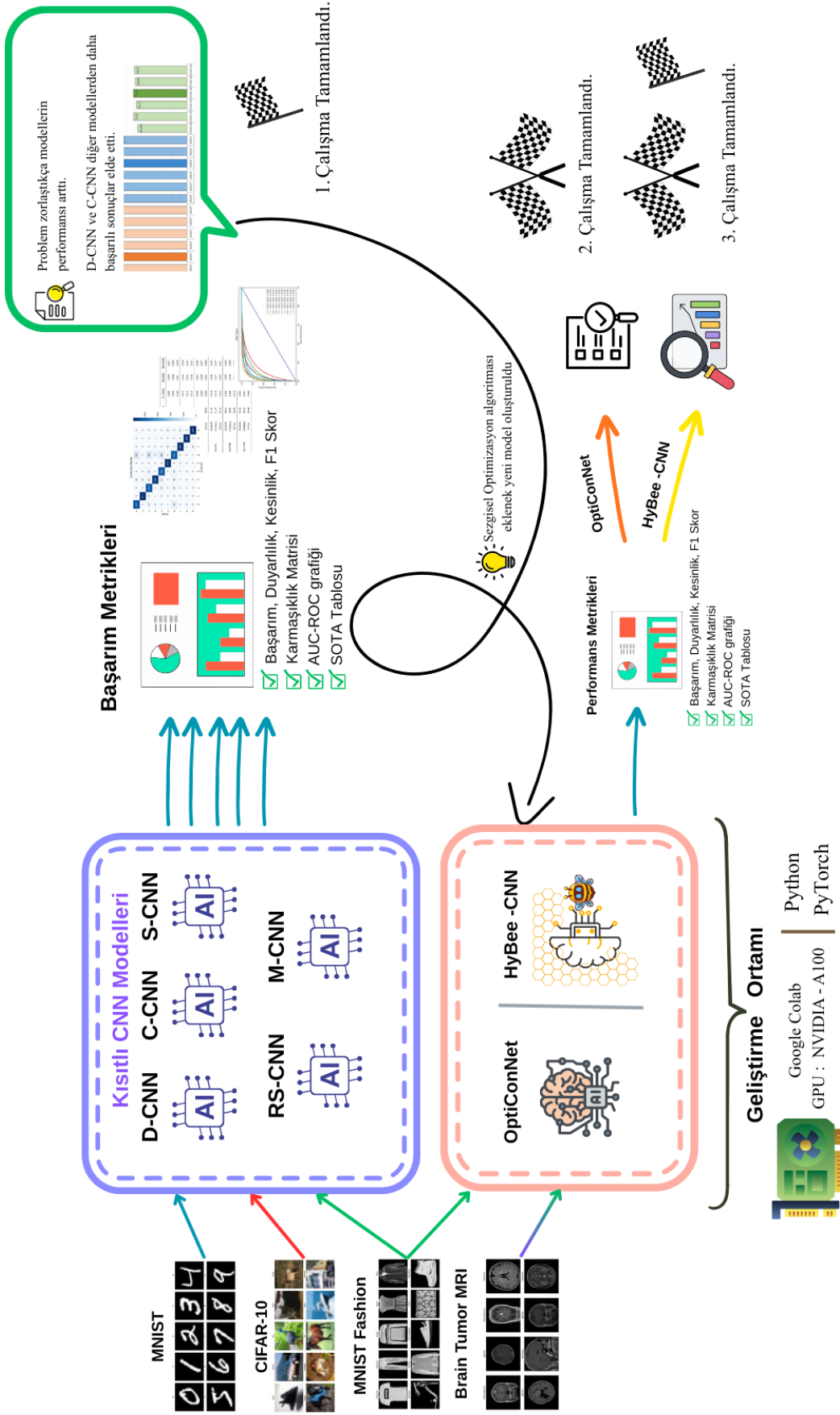
Tablo 1.1'de özetlenen çalışmaların yanı sıra, giriş bölümünde ve deneysel sonuçlar ile tartışma bölümünde aynı veri kümeleri üzerinde gerçekleştirilen diğer araştırmaların sonuçlarına ilişkin karşılaştırmalar yapılmış; bu bağlamda literatür taramasından elde edilen bilgiler diğer bölümler içerisinde de sunulmuştur.

2. METOT

Bu çalışmada, CNN derin ağ mimarisinin başlangıç parametrelerinin çözüm uzayında daha etkin bir başlangıç noktasından başlatılarak model başarımının artırılması hedeflenmiştir. Bu amaçla toplamda yedi farklı model önerilmiştir. Bu modeller;

- D-CNN: Kısıtlı Fark Tabanlı Evrişimli Sinir Ağı (Constrained Difference CNN),
- C-CNN: Örnek Tabanlı Kısıtlamalı Evrişimli Sinir Ağı (Sample Constrained CNN),
- S-CNN: Toplama Tabanlı Evrişimli Sinir Ağı (Constrained Sum CNN),
- RS-CNN: Rastgele Toplam Tabanlı Evrişimli Sinir Ağı (Random Sum CNN),
- M-CNN: Kısıtlı Karma Evrişimli Sinir Ağı (Constrained Mixed CNN),
- OptiConNet: Tek Aday Sınıf Seçimi ile Optimize Edilmiş Karma Kısıtlı CNN Modeli
- HyBee-CNN: Yapay Arı Kolonisi ile Optimizasyonu Gerçekleştirilen Kısıtlı CNN Modeli

Önerilen modeller çeşitli veri kümeleri üzerinde kapsamlı deneysel değerlendirmelere tabi tutulmuştur. Modellerin başarısı yaygın olarak kullanılan başarımların metrikleri aracılığıyla nicel olarak analiz edilmiştir. Çalışmada izlenen genel metodoloji ve aşamalar Şekil 2.1’de gösterilmiştir.



Şekil 2.1 Çalışmanın metodolojik çerçevesi-1.

Şekil 2.1’de özet olarak verilen tez çalışmasının birinci bölümünde, beş farklı Kısıtlı CNN modeli önerilmiştir. Önerilen modellerde, temel CNN mimarisine kısıtlı ağırlık üretici katmanı eklenmiştir. Geliştirilen bu yaklaşımda, ağırlık bağlantı ağırlık parametreleri rastgele başlatılmak yerine, kullanılan veri kümesindeki bilgileri kullanarak başlangıç değerleri belirlenmiş ve eğitim sürecine daha uygun bir noktadan başlanmıştır. Bu sayede, modelin daha etkin bir şekilde eğitilmesi ve sonuçta daha yüksek başarımlı genelleme yeteneği elde edilmesi hedeflenmiştir. Bu beş model, aynı parametreler ile aynı veri kümelerinde eğitilip değerlendirilmiş, eşdeğer yapılandırmaya sahip bir klasik CNN modeliyle karşılaştırmalı analiz edilerek başarımların farkları ortaya konmuştur.

Çalışmanın ikinci bölümünde kısıtlı CNN yöntemleri, sezgisel optimizasyon algoritmalarıyla bütünleştirilerek yeni bir karma modeller oluşturulmuştur. Kısıtlı CNN modellerinin başarımlarını analiz edilerek en yüksek doğruluk elde eden D-CNN ve C-CNN modellerinin fark vektöründen faydalanan kısıtlı ağırlık üretim yaklaşımı ile yeni modeller üretilmiştir. D-CNN ve C-CNN içerisindeki vektör seçim stratejileri yerine optimizasyon algoritmalarından faydalanılarak vektör seçiminin mümkün olduğunca en uygun şekilde yapılması sağlanmıştır. Bu çalışmada optimizasyon sürecinde SCO algoritmasından yararlanılmış ve kısıtlı CNN yapısından türetilen yeni modele “*OptiConNet*” adı verilmiştir. Çalışmanın ilerleyen aşamalarında ise tek çözümlü SCO algoritması, sürü tabanlı ABC algoritması ile değiştirilerek yeni bir karma model olan “*HyBee-CNN*” adı verilen ağ yapısı geliştirilmiştir.

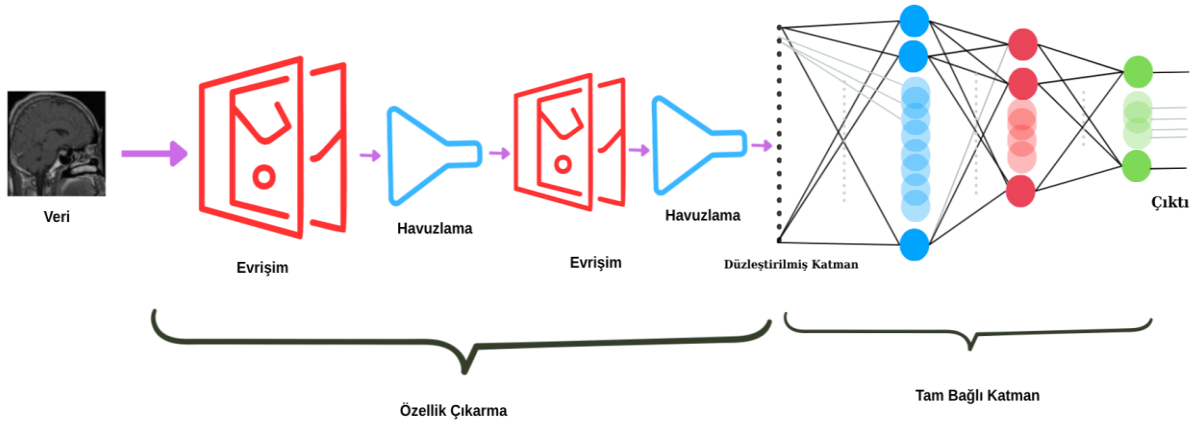
Çalışmada kısıtlı CNN için literatürde kıyaslama için sıklıkla kullanılan MNIST, MNIST Fashion, CIFAR-10 (Canadian Institute for Advanced Research 10) veri kümeleri kullanılmıştır. OptiConNet modeli için ise kıyaslama olarak MNIST Fashion veri kümesi kullanılıp, çalışmanın devamında modelin gerçek hayat problemlerindeki başarımlarını ölçebilmek için Brain Tumor MRI veri kümesi kullanılmıştır. HyBee-CNN modeli için ise MNIST Fashion veri kümesi kullanılmıştır. Yapılan bu uygulamalar doğruluk, kesinlik, duyarlılık, F1 Skor, karmaşıklık matrisi ve AUC-ROC (Area Under the Receiver Operating Characteristic) grafiği gibi metrikler ile değerlendirilmiştir.

Çalışma Google Colab uygulama ortamında, Python programlama dili kullanılarak yapılmıştır. Bu çalışmada, derin öğrenme modellerinin esnek ve etkin bir şekilde uygulanmasına olanak sağlayan ve Facebook AI Research ekibi tarafından geliştirilen PyTorch kütüphanesinden yararlanılmıştır (Ketkar vd. 2021). Eğitim süreci yüksek bellek yoğunluklu

işlemler gerektirdiğinden, Google Colab üzerinde sunulan 24 GB bellek yeteneğine sahip NVIDIA A100 GPU'su kullanılarak model eğitimleri gerçekleştirilmiştir.

2.1. Evrişimli Sinir Ağı (Convolutional Neural Network - CNN)

Evrişimli Sinir Ağı (Convolutional Neural Network – CNN), özellikle görüntü verilerini işlemek, anlamlandırmak veya sınıflandırmak için tasarlanmış bir derin öğrenme mimarisidir (O'shea ve Nash 2015a). Günümüzde görüntü işleme ve video analizi gibi alanlarda yaygın şekilde kullanılmaktadır (Eltoukhy vd. 2025; Wu vd. 2025). CNN temel olarak girdi verisinin yüksek düzeyde soyut ve hiyerarşik özelliklerini otomatik olarak öğrenme yeteneğiyle dikkat çeken bir derin öğrenme yaklaşımıdır (LeCun vd. 2015). Yapısal olarak bir CNN modeli, genellikle iki temel bileşenden oluşur: öznelik çıkarımı için ardışık evrişim (convolution) ve havuzlama (pooling) katmanlarından oluşan ön bölüm ve öğrenilen özelliklerin karar verme amacıyla sınıflandırıldığı, tam bağlı katman (Fully Connected Layer - FCN) bulunmaktadır. Evrişimli bloklarda elde edilen özellik haritaları, genellikle düzleştirilerek tek bir özellik vektörüne dönüştürülür ve bu vektör, çok katmanlı algılayıcı (Multilayer Perceptron, MLP) tabanlı bir sınıflandırıcı tarafından kullanılmaktadır. Böylece ağıın son çıkış katmanında sınıf olasılıkları üretilerek nihai tahmin gerçekleştirilir (Wang vd. 2020). CNN mimarisinin temel bileşenleri Şekil 2.2'de örnek bir şema üzerinden gösterilmiştir.



Şekil 2.2. CNN Mimarisi ve katmanları

Şekil 2.2'de birinci bölüm özellik çıkarma, ikinci bölüm ise tam bağlı katman olarak ayrılmıştır (Wu vd. 2021). Özellik çıkarımı aşamasında, girdi görüntüsü ardışık evrişim, doğrusal olmayan aktivasyon ve alt örnekleme katmanlarından geçirilir, bu süreçte gerekli durumlarda sıfır dolgu (zero-padding) gibi stratejilerle uzamsal boyut korunur. Son evrişim

bloğundan Genişlik \times Yükseklik \times Filtre Sayısını şeklinde üç boyutlu boyutlu aktivasyon tensörü elde edilir. Üç boyutlu olan bu tensör düzleştirme katmanında her bir bloğu arka arkaya dizilerek bir özellik vektörü haline getirilir. Oluşturulan vektör bir sonraki bölüme girdi olarak verilir. İkinci bölümde ise çok katmanlı sinir ağı bulunur. Bu ağ gelen vektör verisini bir girdi olarak alır ve sınıflama işlemi gerçekleştirir. Yapılan sınıflandırma CNN modelinin çıktısını oluşturmaktadır. Eğitim süreci boyunca sınıflandırma işlemini gerçekleştiren CNN modeli, özellik çıkarım aşamasında belirlenen yapılandırma parametrelerine bağlı olarak bir düzleştirme vektörü oluşturur. Eğitim sırasında geriye yayılım algoritması tüm katmanlara uygulanmakta; evrişimli katmanlar ile tam bağlı (MLP) katmanlardaki ağırlıklar öğrenilerek güncellenmektedir. Bu süreçte hem özellik çıkarım katmanları hem de tam bağlı katmanlar ortak bir kayıp fonksiyonuna göre geriye yayılım algoritması ile birlikte güncellenmekte, dolayısıyla modelin tüm eğitilebilir parametreleri eğitim boyunca aktif olarak öğrenilmektedir. Bu öğrenme sürecinin tüm alt adımları aşağıdaki alt başlıklarda verilmiştir.

2.1.1. Özellik çıkarma (Feature extraction)

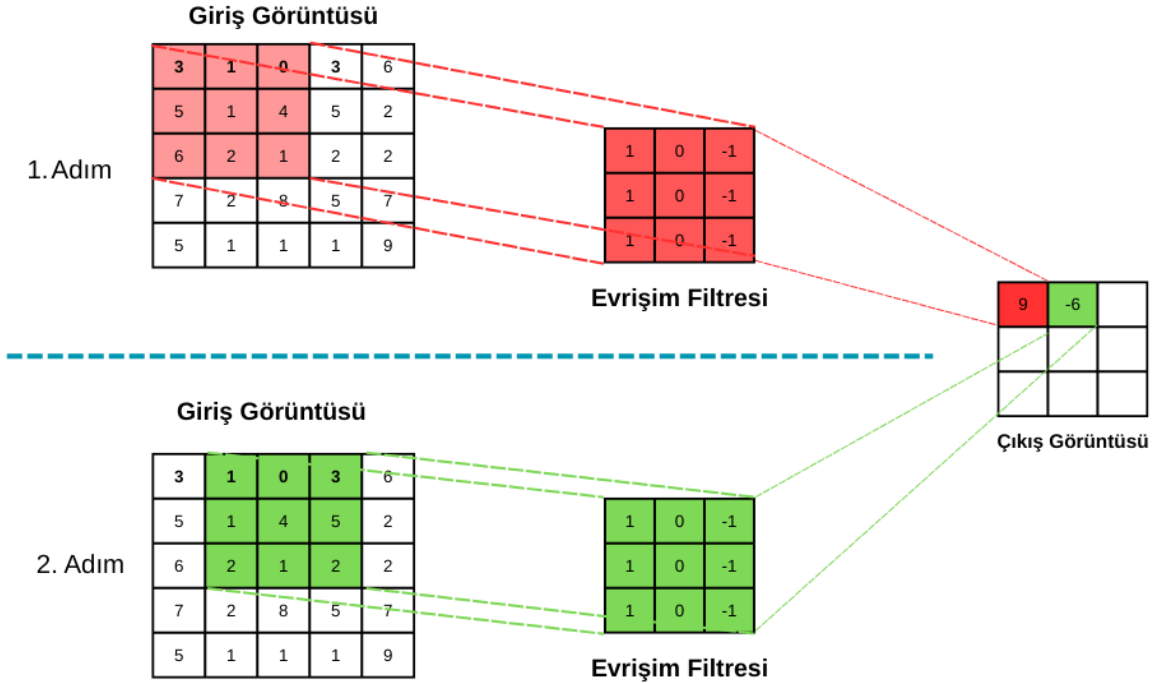
Özellik çıkarma, CNN'nin ilk bölümü olarak kabul edilir. Görsel girdilerin evrişim işlemlerine tabi tutulup dolgulama, havuzlama işlemlerinin yapıldığı ve sonuç olarak görsel içerisindeki özelliklerin özetlenip bir vektör haline getirildiği bölümdür (O'shea ve Nash 2015b). Tezin bu bölümünde CNN'nin içerisinde yapılan evrişim, adım kaydırma, dolgulama ve havuzlama gibi işlemler sırasıyla matematiksel alt yapılarıyla birlikte verilmiştir.

2.1.2. Evrişim katmanı (Convolution layer)

Evrişim işlemi, bir filtre veya çekirdek olarak adlandırılan küçük boyutlu bir matrisin, girdi verisi üzerinde sistematik bir şekilde kaydırılarak uygulanması ile gerçekleştirilir (Chauhan vd. 2018). Bu işlem sırasında filtrenin her bir elemanı, girdi verisindeki karşılık gelen bölgesi ile çarpılır ve elde edilen çarpımların toplamı alınır. Bu toplam, o konumdaki filtrenin tepkisini temsil eder ve tüm konumlar için hesaplanarak özellik haritası oluşturulur. Evrişim işlemi, girdi verisindeki yerel yapıları ve örüntüleri yakalama yeteneğine sahiptir. Bu sayede kenar, köşe, doku özellikleri veya daha karmaşık desenler gibi önemli bilgileri model öğrenebilir (Purwono vd. 2022). Ayrıca evrişim katmanları genellikle adım boyutu ve dolgulama gibi parametrelerle birlikte kullanılarak çıktı boyutu ve sınır etkileri kontrol edilir (Hossain ve Sajib 2019).

Evrişim işlemi ile CNN'ler girdi verisinin yerel özelliklerinden başlayarak daha derin

katmanlarda giderek karmaşık ve yüksek seviyeli örüntüleri öğrenebilir (Alzubaidi vd. 2021b). Evrişim katmanlarının birden fazla filtre ve aktivasyon fonksiyonu ile birlikte kullanılması, CNN'in güçlü bir temsil yeteneği kazanmasını sağlar. Evrişim işleminin hesaplama alt yapısını gösteren basit bir örneği Şekil 2.3'te iki adımlı olarak gösterilmiştir.



Şekil 2.3. Evrişim işlemi.

Şekil 2.3'te birinci adımda, 3×3 boyutundaki evrişim filtresi, girdi görüntüsündeki kırmızı ile vurgulanan bölüm ile örtüştüğü girdi görüntüsü piksellerinin eleman bazında çarpımı gerçekleştirilir. Çarpım sonuçları toplandıktan sonra elde edilen skaler değer, çıktı özellik haritasında kırmızı ile gösterilen alana yerleştirilir. Hesaplanan işlemin bir sonraki aşamasında filtrenin bir adım sağa kaydırılarak yapılan hesaplama yeşil dolgu ile verilmiştir. Şekil 2.3'te anlatılan bu işlemin matematiksel ifadesi için girdi görüntüsü $I \in \mathbb{R}^{H \times W}$ (yükseklik \times genişlik) ve evrişim filtresi $K \in \mathbb{R}^{k \times k}$ olmak üzere, çıktı özellik haritası $O \in \mathbb{R}^{H' \times W'}$ olacak şekilde Eşitlik D.1'de verilen formül ile evrişim katmanı çıkışı hesaplanır.

$$o(i, j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I(i + m - p, j + n - p) \cdot K(m, n) + b \quad (D.1)$$

Eşitlik D.1'de i, j çıktı özellik haritasındaki piksel koordinatları, p dolgulama, b eşik

değerini (bias) ifade etmektedir. Çıkış görüntüsünün matris boyutları hesaplanırken dolgulama ve adım parametreleri çıkış matrisinin boyutlarını etkilemektedir. Çıkış görüntüsünün boyut hesaplaması Eşitlik D.2 ile yapılmaktadır.

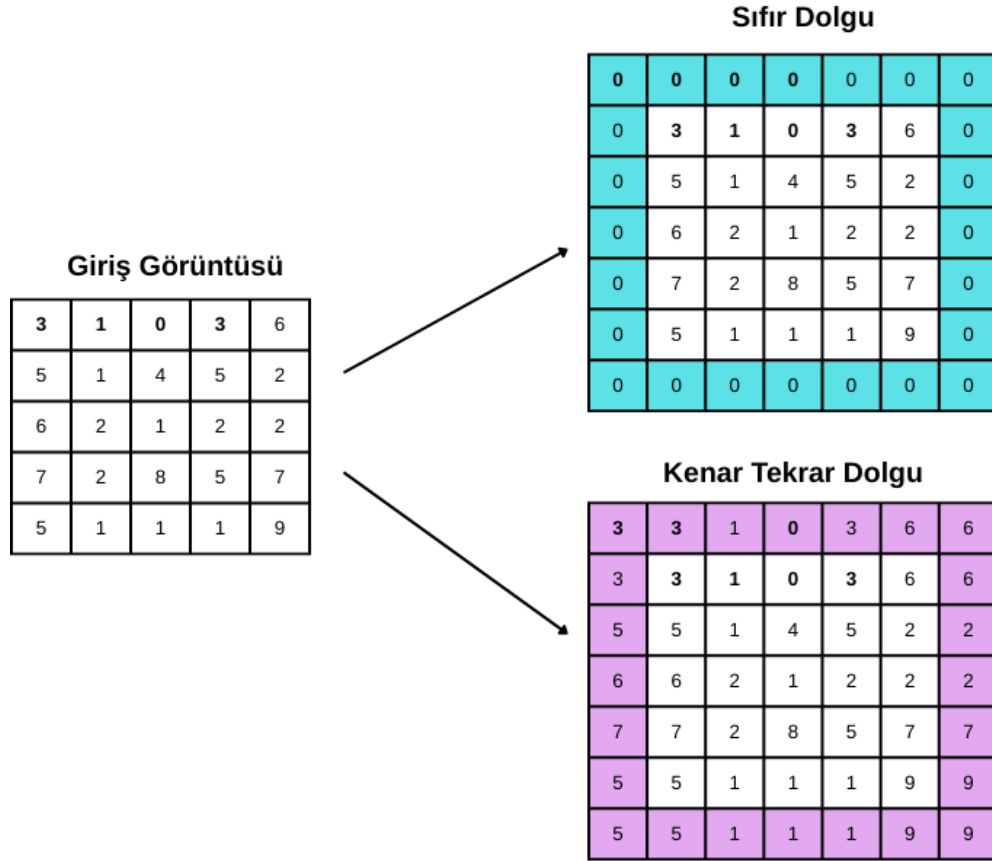
$$H' = \left\lceil \frac{H + 2p - k}{s} \right\rceil + 1, \quad W' = \left\lceil \frac{W + 2p - k}{s} \right\rceil + 1 \quad (\text{D.2})$$

Eşitlik D.2'de hesaplanan $H' \times W'$ çıktı görüntüsünün boyutlarını, H girdi görüntüsünün yüksekliğini, W girdi görüntüsünün genişliğini, k filtre boyutunu, s ise adım kaydırma miktarını ifade etmektedir.

2.1.3. Dolgulama (Padding)

Evrişim işlemi sırasında çıkış öznitelik haritalarının, çekirdek boyutundan kaynaklanan hesaplama etkisiyle küçülmesi, özellikle derin ağlarda uzamsal çözünürlüğün erken aşamalarda azalmasına ve kenarlardaki bilginin kaybolmasına yol açtığından genellikle istenmeyen bir durumdur. Bu sorunu azaltmak amacıyla, girdi matrisinin sınırlarına ek pikseller eklenerek dolgulama uygulanır ve böylece hem uzamsal boyut korunur hem de kenar bilgisi kaybı minimuma indirilir (Naseri ve Mehrdad 2023). Dolgu işleminin temel amacı, filtre merkezinin (yani evrişim işleminin referans noktası olarak alınan pozisyonun) girdinin kenar bölgelerindeki piksellerle hizalanabilmesini sağlamaktır. Matematiksel olarak, bir $k \times k$ boyutundaki filtrenin merkez indeksi ($[k/2], [k/2]$) olarak tanımlanır ve bu noktanın girdi matrisinin (0,0) koordinatındaki piksele hizalanabilmesi için simetrik bir dolgu uygulanması gerekir. Burada en sık kullanılan iki farklı dolgu stratejisi sunulmaktadır. Bunlardan ilki olan Sıfır Dolgu (Zero Padding) işleminde girdi matrisinin çevresine belirlenen miktarda sıfır değerli pikseller ilave edilir. Bu yöntem hesaplamada ek veri üretmeden çıktı boyutunun istenen düzeyde tutulmasını sağlar ve eğitim verimliliği açısından tercih edilir (Cheng 2020). İkinci yöntem ise Kenar Tekrarlama Dolgusunda (Replication Padding), girdinin dış sınırlarına komşu kenar piksellerinin değerleri doğrudan kopyalanarak dolgu oluşturulur (Liu vd. 2018). Örneğin, bir piksel genişliğinde sol kenar dolgusu eklenirken, orijinal görüntünün ilk sütunundaki pikseller yeni sütuna sırayla aktarılır; aynı işlem sağ, üst ve alt kenarlar için de simetrik olarak uygulanır. Birden fazla piksel dolgusu gerektiren durumlarda, her yeni katman dıştan içe doğru yalnızca bir önceki sınır satır/sütunundan alınan değerlerle doldurulur, yani iç bölgelerden hiyerarşik

kopyalama yapılmaz. Bu strateji, kenarlarda yapay geçişlerin (border artifacts) azaltılmasına ve yerel doku sürekliliğinin korunmasına katkıda bulunabilir. Şekil 2.4'te, evrişim işleminde sunulan örnek girdi görüntüsüne sırasıyla sıfır dolgu ve kenar tekrarlama dolgusu uygulanmış örnekler gösterilmiştir.



Şekil 2.4 Sıfır dolgu ve kenar tekrar dolgu örnekleri.

2.1.4. Adım kaydırma (Stride)

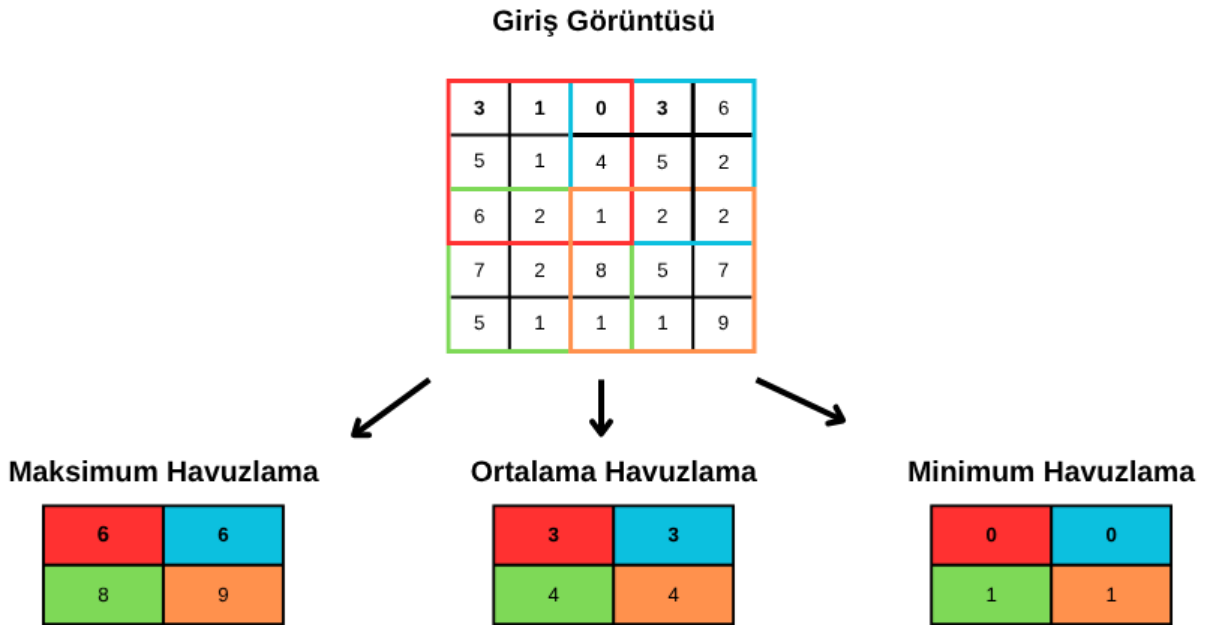
Evrişim işlemi sırasında kullanılan filtre matrisinin, giriş görüntüsü üzerinde bir konumdan bir sonrakine geçerken yatay veya dikey ekseninde kaç piksel ilerleyeceği adım kaydırma (stride) ile belirlenir. Bu işlem filtrenin görüntü üzerinde tarama hızını ve örnekleme yoğunluğunu kontrol eden temel bir parametredir (Guo vd. 2019).

Adım kaydırma değeri küçük olduğunda, filtre görüntü üzerinde daha sık bir şekilde kaydırılır ve bu durum daha ayrıntılı bir özellik çıkarımı sağlar. Sonuç olarak çıkış matrisi daha büyük olur ve modelin hesaplama yükü artar. Adım kaydırma değeri büyütüldüğünde filtre daha geniş aralıklarla hareket eder, böylece çıkış matrisi küçülür, hesaplama maliyeti azalır ve model daha hızlı çalışabilir. Adım kaydırma değerini fazla büyük seçilmesi, görüntüdeki ince

ayrıntılarının atlanmasına, dolayısıyla hassasiyet kaybına ve bazı kritik özelliklerin kaçırılmasına yol açabilir. Bu nedenle adım kaydırma değeri, çözülmek istenen probleme, giriş görüntülerinin çözünürlüğüne ve modelin hedeflenen karmaşıklık seviyesine göre dikkatle seçilmelidir. Yüksek çözünürlüklü ve detaylı görüntülerde daha küçük bir adım kaydırma tercih edilirken, düşük çözünürlüklü veya daha karmaşık olmayan örüntülerin çıkarılmasının yeterli olduğu durumlarda daha büyük bir adım kaydırma kullanılabilir.

2.1.5. Havuzlama (Pooling)

Havuzlama (Pooling) özellik haritalarının boyutunu azaltmak, örnekleme yapmak ve ağız işlem yükünü azaltmak için kullanılır. Genellikle evrişim katmanından hemen sonra kullanılır (Zafar vd. 2022a). Havuzlama işleminde ilgili bölgedeki piksel değerleri özetlenerek boyutları küçültülür ve bu özet bilgi bir sonraki katmana aktarılır. Maksimum havuzlama, ortalama havuzlama ve minimum havuzlama şeklinde üç farklı yöntem ile uygulanabilir (Sun vd. 2017). Maksimum havuzlama işlemi ilgili havuz alanının içindeki en yüksek piksel değerini alır, minimum havuzlamada bu alandaki minimum değer alınırken, ortalama havuzlama katmanında ise alandaki piksellerin aritmetik ortalaması alınmaktadır. Şekil 2.5'te üç havuzlama yöntemine dair bir örnek hesap gösterilmektedir.



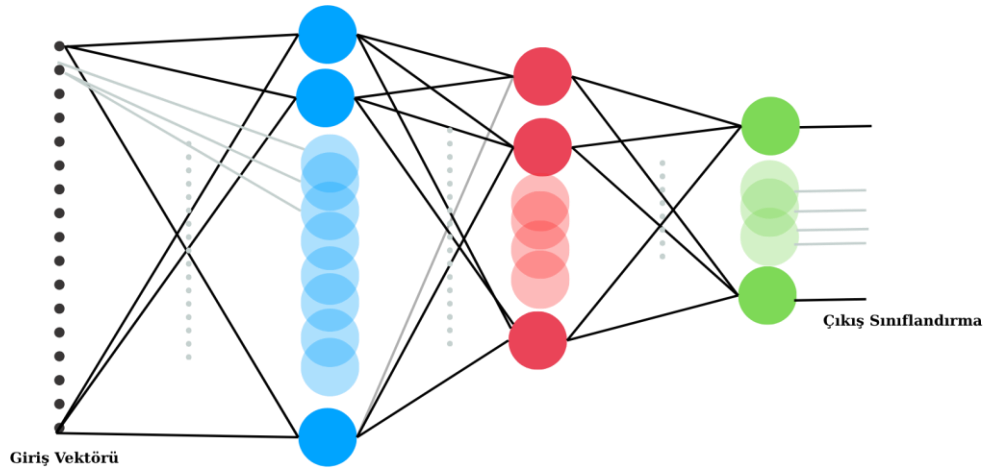
Şekil 2.5. Havuzlama yöntemlerine ait uygulama örnekleri.

Sonuç olarak havuzlama ile ilgili bölgeden bir örneklem alınarak hesaplama yükü hafifletilir. Ayrıca çalışma kapsamında yapılan incelemelerde literatürde çoğunlukla maksimum havuzlama kullanıldığı, bazı çalışmalarda ise ortalama havuzlama kullanıldığı en

az minimum havuzlama kullanıldığı gözlemlenmiştir (Ajit vd. 2020; Gholamalinezhad ve Khosravi 2020; Zafar vd. 2022b).

2.1.6. Tam bağlı katman (Fully connected network – FCN)

CNN mimarisinin ikinci bölümü olarak görülür ve evrişim işlemlerinden sonra sınıflama yapmak için kullanılan katmandır (Basha vd. 2020). Klasik CNN yönteminde bu katmanda MLP modeli kullanılır. MLP haricinde farklı yöntemlerin uygulandığı çalışmalar literatürde bulunmaktadır (Duan vd. 2018). Klasik yöntemdeki bu MLP ağı geriye yayılım yöntemi ile hesaplanıp eğitilir. Şekil 2.6’da tam bağlı katman örneği verilmiştir.



Şekil 2.6 Tam bağlı katman örneği.

Şekil 2.6’da giriş vektöründeki her bir girdiden her bir giriş katmanı hücresine bir bağlantı bulunmaktadır. Bu bağlantıların ağırlık olarak belirtilen bir çarpan değeri bulunmaktadır. Her bir hücrenin çıkışı hesaplanırken ona gelen tüm girişler bağlı olduğu bağlantı ağırlığı ile çarpılıp elde edilen sonuçlar toplanır. Elde edilen toplam değer bir aktivasyon fonksiyonundan geçilip katmanın çıktısı hesaplanır. Birinci katmanın çıktısı Eşitlik D.3 ile hesaplanır.

$$Z^{(l)} = \sigma^{(l)} (W^{(l)} \times X + b^{(l)}) \quad (D.3)$$

Burada X giriş vektörünü, $W^{(1)}$ birinci katmanın bağlantı ağırlık matrisini, $\sigma^{(1)}$ birinci katmanın aktivasyon fonksiyonunu, $b^{(1)}$ birinci katmandaki tüm nöronlara giden eşik (bias)

vektörünü ve $Z^{(1)}$ birinci katmanın çıktısını temsil etmektedir. İkinci ve sonraki katman çıkışlarını hesaplaması için Eşitlik D.4 uygulanır.

$$Z^{(l)} = \sigma^{(l)} (W^{(l)} \odot Z^{(l-1)} + b^{(l)}), \quad l = 2,3, \dots, L \quad (D.4)$$

Eşitlik D.4'te \odot işleci eleman bazlı çarpımı ifade etmektedir. Eşitlik D.4 bir önceki Eşitlik D.3'teki parametrelere benzer parametreler içermektedir ancak girdi yerine bir önceki katmanın çıkışı ($Z^{(l-1)}$) kullanılır. Böylelikle ağın derinliği artırılarak hiyerarşik bir hesaplama mümkün olur. Çıkış katmanına kadar her bir katman için Eşitlik D.4 kullanılır. Çıkış katmanı hesaplandığında ileri hesap tamamlanmış olur. İleri hesaptan elde edilen çıkış değeri ile gerçek değer arasındaki fark kayıp/hata olarak ifade edilir. Oluşan kayıp için geriye doğru her ağırlığın hata üzerindeki etkisi hesaplanır. Yerel gradyen hesaplama işlemi çıkış katmanından başlayarak, giriş katmanı sıralamasıyla Eşitlik D.5 ve Eşitlik D.6'da verilmiştir.

$$\delta_2 = (y - \gamma) \times f'(Z^{(2)}) \quad (D.5)$$

$$\delta_1 = \delta_2 \times W^{(2)} \times f'(Z^{(1)}) \quad (D.6)$$

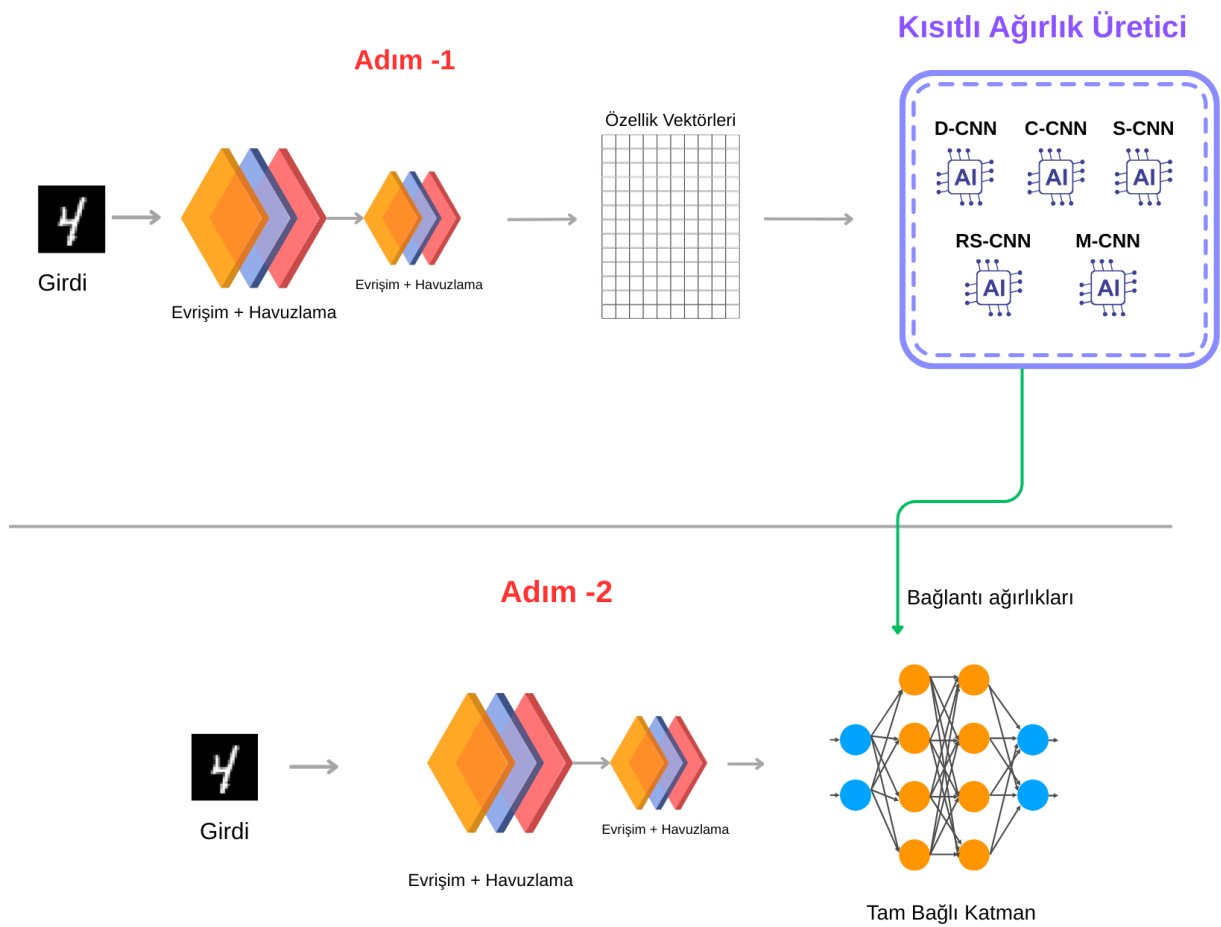
Eşitlik D.5 ve Eşitlik D.6'da f' kativasyon işleminin türevinin alınmasını, γ ağın gerçek hedef çıktısı, y ağın tahmin ettiği çıktı ve δ ise yerel gradyeni ifade etmektedir. Burada hesaplanan hata sinyalleri bir sonraki eşitlikte ağırlıkları güncellemek için kullanılacaktır. Bağlantı ağırlıklarının güncellemesi için Eşitlik D.7 uygulanır.

$$W_{i+1} = W_i - \eta \times (\delta_1 \times X^T) \quad (D.7)$$

Eşitlik D.7'de η öğrenme oranını, $(\delta_1 \cdot X^T)$ girişe bağlı üretilen gradyene, δ yerel gradyeni ve W bağlantı ağırlıklarını ifade eder. Eşitlik D.7'de hesaplanan bağlantı ağırlıkları bir sonraki iterasyonda yeni bağlantı ağırlıkları olarak kullanılır.

2.2. Kısıtlı Evrişimli Sinir Ağları (Constrained CNN)

Kısıtlı CNN, klasik CNN yapısına, kısıtlı ağırlık üretim süreci entegre edilerek oluşturulmuş iki aşamalı bir karma modeldir. Önerilen model özellikle tam bağlantılı katmanda ağırlıkların rastgele başlatılmasının yerine, kısıtlı ağırlık üretici (Constrained Weight Generator) aracılığıyla önceden optimize edilmiş bağlantı ağırlıklarının kullanılmasını sağlamaktadır. Ağırlıkların tamamen rastgele noktalar yerine, veri kümesinin özelliklerine daha uygun başlangıç değerleriyle başlatılması, öğrenme sürecinde hem yakınsama hızının artmasına hem de sınıflandırma doğruluğunun iyileşmesine katkı sağlamaktadır. Modelin genel akışına ait öbek şema Şekil 2.7’de gösterilmiştir.



Şekil 2.7 Kısıtlı CNN işlem akış öbek şeması.

Şekil 2.7’de mimari çizimi sunulan model iki adımdan oluşmaktadır. Birinci adımda, yalnızca öznelik çıkarım katmanlarından oluşan bir Evrişimli Sinir Ağı (CNN) ile bunu izleyen bir kısıtlı ağırlık üretici yer almaktadır. İkinci adımda ise geleneksel bir CNN mimarisi uygulanmaktadır. Söz konusu şematik gösterim, önerilecek beş farklı kısıtlı modelin temel

yapısını örneklemek amacıyla hazırlanmıştır. Önerilen her bir modelde, kısıtlı ağırlık üreticinin iç algoritması ve hesaplama yöntemi birbirinden farklılık göstermektedir.

Modelin birinci adımı klasik CNN mimarisinin yalnızca özellik çıkarım sürecinin yürütüldüğü kısımdır ve sonrasında bağlantı ağırlıklarının üretildiği bölümdür. Bu bölümde yapılan işlemler sırasıyla:

- Giriş veri kümesi, ardışık evrişim ve havuzlama katmanlarından geçirilerek çok katmanlı bir özellik çıkarım işlemi uygulanır.
- Evrişim katmanlarında filtreler, girdi üzerinde kaydırmalı evrişim operasyonları gerçekleştirir, bu işlemleri izleyen havuzlama adımı ile veri özetlenir.
- Havuzlama sonucunda elde edilen çıktılar düzleştirme katmanına aktarılır ve her bir örnek için sabit boyutlu bir özellik vektörü oluşturulur.
- Üretilen bu özellik vektörleri, modelin sonraki fazında kullanılmak üzere bellekte saklanır ve kısıtlı ağırlık üreticiler tarafından işlenmeye hazır hâle getirilir.
- Tüm veri kümesinden istenilen özellik vektörleri elde edilip bağlantı ağırlıklarının hesaplaması yapılır. Bu kapsamda, özellik vektörlerinden ilgili kısıtlı modelin algoritmasındaki kurallarına uygun biçimde vektör seçimi gerçekleştirilir.
- Seçilen iki özellik vektörü, modele bağlı olarak toplam ya da fark vektörü hesaplanır ve bu hesaplanan vektörler aracılığıyla tam bağlantılı katmandaki her bir nörona ait giriş ağırlıkları ve eşik değerleri belirlenir. Bu işlem nöron sayısı kadar tekrar edilerek ağırlıklar ve eşikler hesaplanır.

Böylelikle birinci bölümdeki işlemler tamamlanarak kısıtlı ağırlıklar ve eşik değerleri üretilmiş olur. Burada hesaplanan ağırlıklar ve eşik değerleri ikinci bölümdeki eğitim için kullanılır.

Modelin ikinci adımında, klasik CNN eğitiminin gerçekleştirildiği aşamadır. Bu adımda birinci adımda elde edilen kısıtlı bağlantı ağırlıkları, ağırlık tam bağlantılı katman için başlangıç parametreleri olarak atanır. Böylece model, rastgele ağırlıklarla değil, önceden belirlenmiş ağırlık ve eşik değerlerinden oluşan bir kümeyle eğitime başlar. Bu durum, ağırlık kaybı fonksiyonunu daha düşük bir değerden başlatmasını sağlar. Böylece ağırlık başarımı artırılmış

olur. Bu iki aşamalı yapı sayesinde model, klasik CNN'lerin öğrenme sürecinde rastgele başlatmadan kaynaklanan kararsızlıkları azaltmakta ve daha kararlı bir yakınsama davranışı sergilemektedir. Özellikle problemin zor olduğu veri kümelerinde, kısıtlı ağırlık üreticiler tarafından sağlanan optimize başlangıç değerleri, modelin genel doğruluk oranını artırmaktadır. Önerilen bu mimari, derin öğrenme tabanlı sınıflandırma problemlerinde daha dengeli bir öğrenme süreci ve daha yüksek başarı oranı sunmaktadır.

Kısıtlı CNN yaklaşımı kapsamında, genel çalışma yöntemi iki aşamalı olarak tanımlanan bu karma mimariye dayanarak, vektör seçimi ve seçilen vektörler üzerinde gerçekleştirilecek aritmetik işlemler bakımından birbirinden farklı yöntemleri benimseyen D-CNN, C-CNN, S-CNN, RS-CNN, M-CNN modelleri önerilmiştir. Önerilen modelleri birbirinden ayıran vektör seçimi ve hesaplama yöntemi Tablo 2.1'de verilmiştir.

Tablo 2.1 Kısıtlı CNN modelleri çalışma yöntemleri özet tablo.

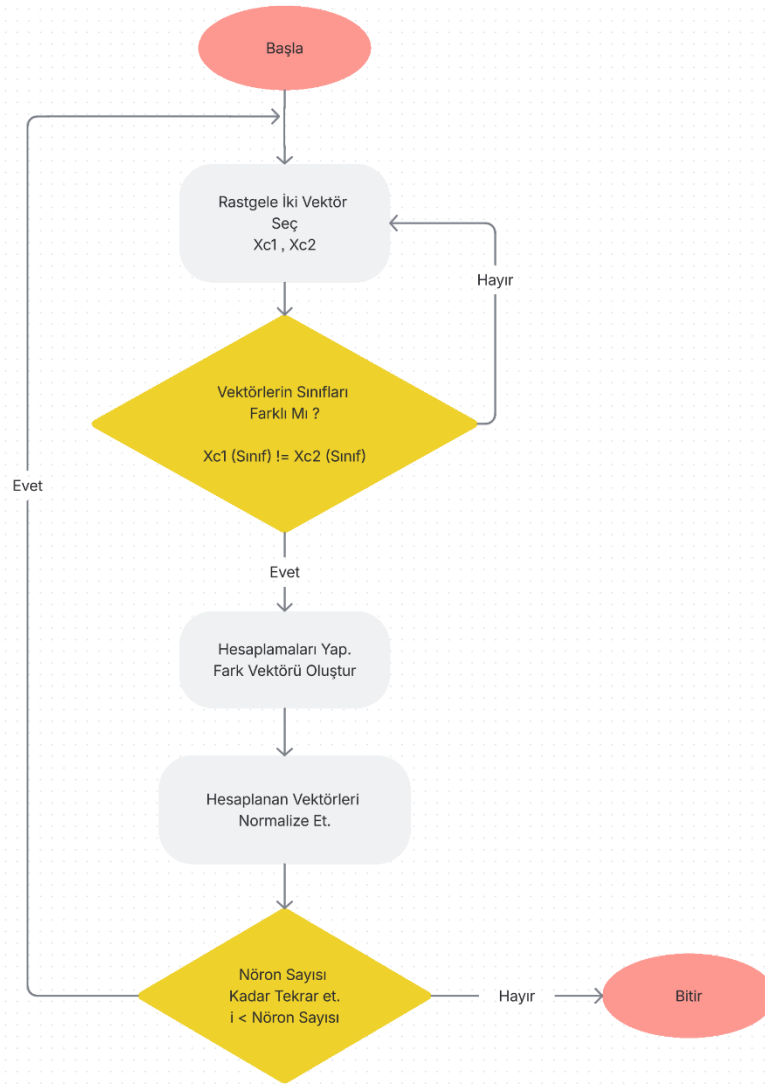
Model	Vektör Seçimi	Hesaplama Yöntemi
D-CNN	Farklı sınıflar	Fark Vektörü
C-CNN	Aynı sınıflar	Fark Vektörü
S-CNN	Aynı sınıflar	Toplam Vektörü
RS-CNN	Rastgele	Toplam Vektörü
M-CNN	Aynı / Farklı sınıflar	Fark / Toplam vektörü

Tablo 2.1 incelendiğinde iki tane fark vektörü hesaplamasını temel alan, iki tane de toplam vektörü hesaplamasını temel alan model bulunmaktadır. Ayrıca iki yöntemi birleştirip karma yapı oluşturan her iki yaklaşımı da içeren M-CNN isimli bir model daha bulunmaktadır. Modeller arasındaki temel hesaplama farklılıkları bu tabloda gösterilmiştir. Ayrıca, bölümün ilerleyen kısmında önerilen bu beş model ayrıntılı olarak açıklanmış; her birinin çalışma ilkesi, akış şeması ve sözde kod (pseudocode) temelli algoritması sunulmuştur.

2.2.1. Kısıtlı fark tabanlı evrişimli sinir ağı (D-CNN)

D-CNN yönteminde, ilk adım olarak görüntüler evrişim işlemlerinden geçirilir ve bu işlem sonucunda yüksek boyutlu özellik vektörleri elde edilir. Elde edilen bu vektörler kısıtlı

ağırlık üretici katmanına verilir. Kısıtlı ağırlık üretici katmanında hesaplamalar bu vektörler üzerinden yapılır. D-CNN modelinde bulunan kısıtlı ağırlık üreticisi ilk olarak farklı sınıflara ait rastgele iki vektör seçer ve bu vektörler bir birinden çıkartılır. Elde edilen fark vektörü normalize edilir. Normalize edilen vektör tüm girişlerden FCN katmanındaki birinci nörona giden bağlantı ağırlıklarının başlangıç parametrelerini oluşturur. Bu işlem nöron sayısı kadar tekrarlanarak ağırlık üretici katmanın tüm parametreleri elde edilerek kısıtlı ağırlıklar ve eşik değerlerinin bağlantı ağırlıkları üretilmiş olur. Yapılan bu hesaplama yönteminin akış şeması Şekil 2.8’de verilmiştir.



Şekil 2.8 D-CNN ağındaki tam bağlı katman ağırlıkları ve eşik parametreleri belirleme işlemleri akış şeması.

Şekil 2.8’de verilen akış şeması kısıtlı CNN modellerinden D-CNN’e özgü hesaplamaların yapıldığı kısıtlı ağırlık üretici katmanının çalışma akışı çizilmiştir. Akış

şemasında dikkat çekici iki nokta bulunmaktadır. Birincisi algoritmada seçilen vektörlerin sınıflarının farklı olmasına zorlanmaktadır. İkincisi ise vektörler arasında yapılan aritmetik çıkarma işleminin bulunmasıdır. D-CNN için geliştirilen bu yapının sözde kodu Algoritma 2.1’de verilmiştir.

Algoritma 2.1: D-CNN modelinin sözde kodu

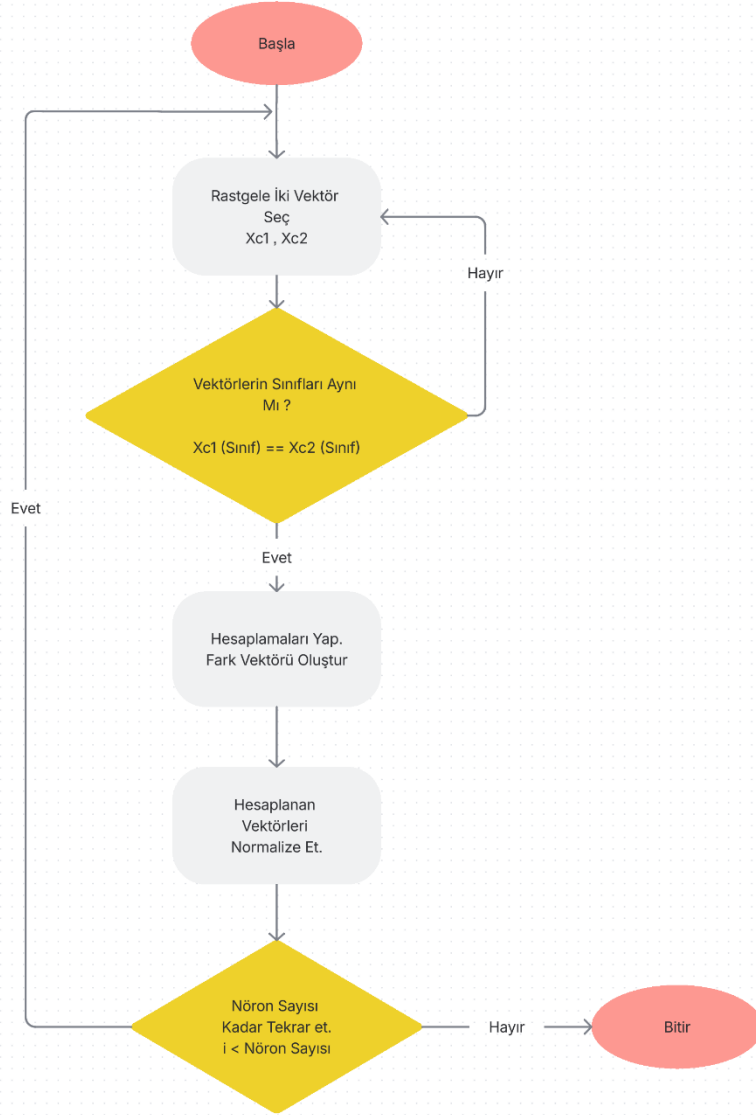
- 1) Veri kümesi konvolüsyon işleminden geçirilir ve düzleştirme katmanından vektör elde edilir.
 - 2) **for** $i = 1:N$ **do**
 - a) Rastgele olarak farklı sınıflara ait iki vektör X_{c1} ve X_{c2} seçilir
 - b) Fark vektörü $X_{c2} - X_{c1}$ oluşturulur
 - c) Fark vektörü normleştirilerek bulunur: $W_i = \frac{(X_{c2} - X_{c1})}{\|X_{c2} - X_{c1}\|}$
 - d) Eşik değeri hesaplanır: $b_i = \frac{(X_{c1} + X_{c2})^T (X_{c1} + X_{c2})}{\|X_{c2} - X_{c1}\|}$
 - e) Elde edilen w_i ve b_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı ve $b_{i \times L}$ eşik değeri elde edilir
 - 3) **end for**
 - 4) Elde edilen W matrisi ve b eşik vektörü CNN modelinin FCN katmanının başlangıç değerleri olarak atanır.
 - 5) CNN modelinin eğitim süreci başlatılır.
-

Algoritma 1’de sözde kodu verilen D-CNN modeli özetle farklı sınıflara ait vektörlerin farklarından yararlanarak klasik CNN mimarilerinden farklı olarak tam bağlantılı katman ağırlıklarının rastgele değil, girdi vektörlerinin arasındaki ilişkiden faydalanarak modelin daha uygun bir noktadan eğitime başlamasını sağlamaktadır. Bu model ile beraber aritmetik işlemleri ve sınıf seçimleri birinden farklı olan dört model daha önerilmiştir.

2.2.2. Örnek tabanlı kısıtlamalı evrişimli sinir ağı (C-CNN)

C-CNN yöntemi, D-CNN modeline benzer şekilde fark vektörlerinden yararlanarak bağlantı ağırlıklarının başlangıç parametrelerini belirli bir uygun değerden başlatmayı amaçlayan bir modeldir. Model rastgele seçilen vektörlerin aynı sınıfa ait olmasını zorunlu kılar ve vektörlerin farkını hesaplar. D-CNN modelinde vektörlerin sınıfları farklı olarak seçilirken C-CNN modelinde aynı sınıf olacak şekilde seçim yapılır. İki model arasındaki en önemli fark

budur. Elde edilen fark vektörü normalize edilir. Bu işlemler nöron sayısı kadar tekrarlanarak başlangıç matrisi oluşturulur. C-CNN ait bu çalışma ilkesinin akış şeması Şekil 2.9'da sunulmuştur.



Şekil 2.9 C-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması.

C-CNN modelinin FCN ağırlıklarını hesaplama süreci Şekil 2.9'da verilen akış şemasında özetlenmiştir. Akış şemasında dikkat çeken temel unsur vektörler seçilirken C-CNN'de aynı sınıfa ait vektörler seçilmektedir. C-CNN modeli uygulanırken kullanılan sözde kodu Algoritma 2.2'de verilmiştir.

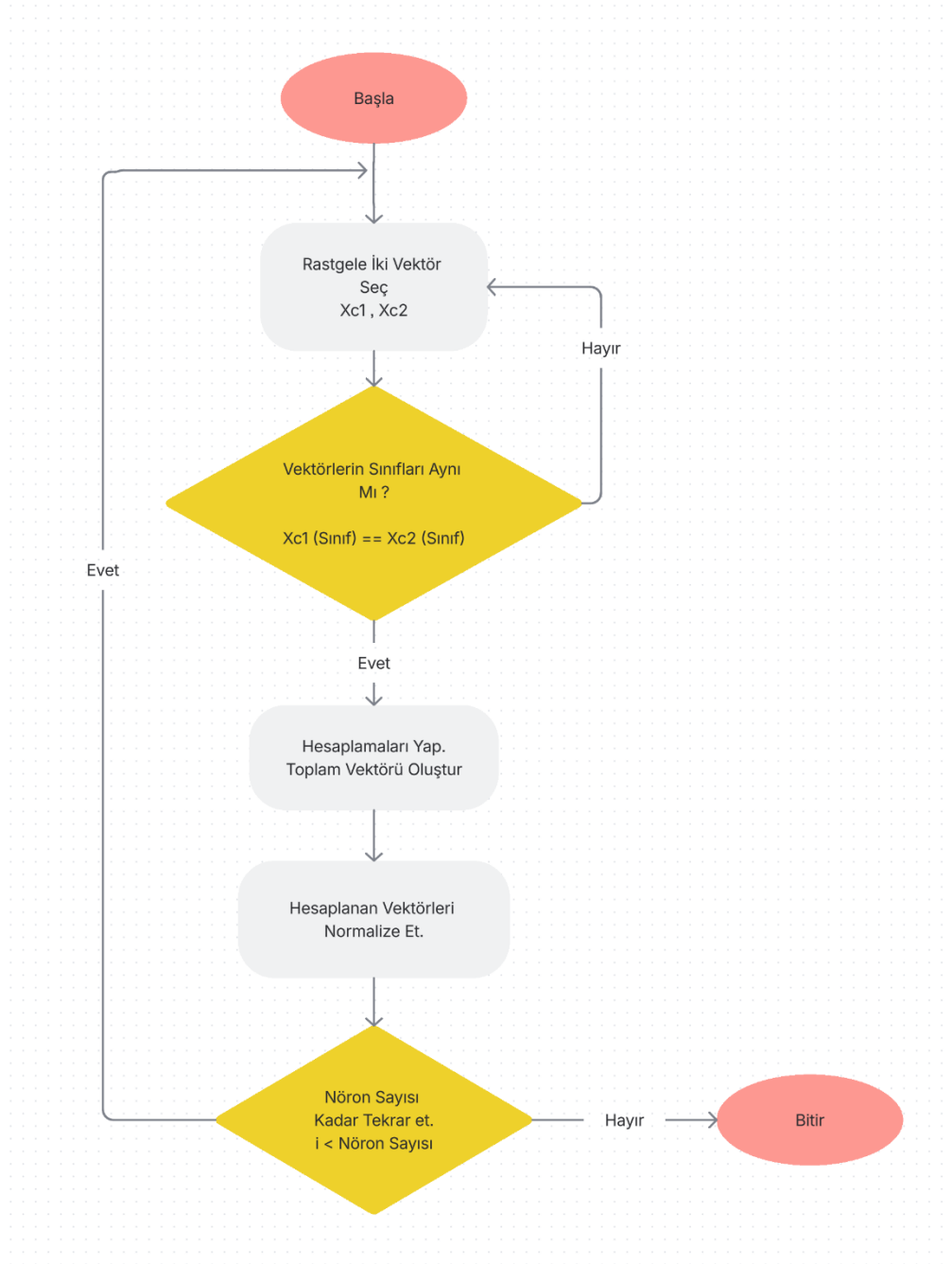
Algoritma 2.2: C-CNN modelinin sözde kodu

- 1) Veri kümesi konvolüsyon işleminden geçirilir ve düzleştirme katmanından veri elde edilir.
 - 2) **for** $i = 1:N$ **do**
 - a) Rastgele olarak aynı sınıftan iki vektör X_{c1} ve X_{c2} seçilir
 - b) Fark vektörü $X_{c2} - X_{c1}$ oluşturulur
 - c) Fark vektörü normalleştirilerek bulunur: $W_i = \frac{(X_{c2} - X_{c1})}{\|X_{c2} - X_{c1}\|}$
 - d) Elde edilen w_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı elde edilir.
 - 3) **end for**
 - 4) Elde edilen W matrisi CNN modelinin FCN katmanının başlangıç değerleri olarak atanır.
 - 5) CNN modelinin eğitim süreci başlatılır.
-

Sonuç olarak C-CNN modeli, sınıf içi benzerlikleri de kullanarak öğrenme sürecinin başlangıcında modelin daha iyi bir eğitim sürecini sağlayarak sonuçlarda elde edilen başarımın daha iyi çıkmasını sağlamaktadır.

2.2.3. Toplama tabanlı evrişimli sinir ağı (S-CNN)

S-CNN modelinde fark işlemi yerine vektörlerin toplamından faydalanarak ağırlık üretilir. Vektör seçimi yapılırken aynı sınıfa ait vektörler seçilerek hesaplama yapılır. S-CNN modelinin akışı şeması Şekil 2.10'da şema olarak verilmiştir.



Şekil 2.10 S-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması.

S-CNN modelinin FCN katmanı parametrelerinin ilk atama çalışma süreci Şekil 2.10’da verilen akış şemasında gösterilmiştir. Akış şemasında aynı sınıfların seçilmeye zorlanması C-CNN ile benzer bir akışa sahip olduğunu göstermektedir ancak hesaplama kısmında ise fark vektörü yerine toplam vektörü elde edilmesi ile fark vektörü temelli algoritmalarından ayrılmaktadır. Bu toplam vektörünü bulma işlemi ile aynı sınıfa ait örneklerin benzer özelliklerini bir araya getirerek modelin bu ortak örüntüleri daha güçlü bir biçimde temsil etmesini sağlamaktadır. S-CNN algoritmasının sözde kodu Algoritma 2.3’te verilmiştir.

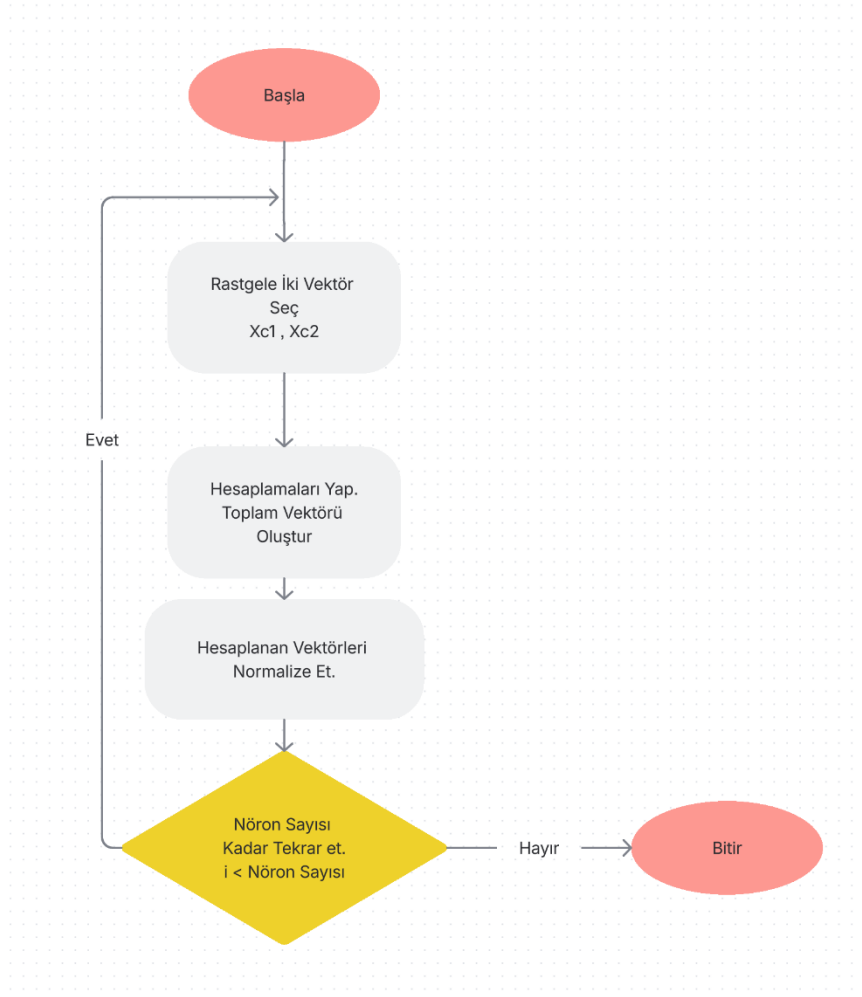
Algoritma 2.3: S-CNN modelinin sözde kodu

- 1) Veri kümesi konvolüsyon işleminden geçirilir ve düzleştirme katmanından veri elde edilir.
 - 2) **for** $i = 1:N$ **do**
 - a) Rastgele olarak aynı iki sınıftan iki vektör X_{c1} ve X_{c2} seçilir.
 - b) Toplam vektörü $X_{c2} + X_{c1}$ oluşturulur.
 - c) Toplam vektörü normalleştirilerek bulunur: $W_i = \frac{(X_{c2} - X_{c1})}{\|X_{c2} + X_{c1}\|}$
 - d) Elde edilen w_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı elde edilir.
 - 3) **end for**
 - 4) Elde edilen W matrisi CNN modelinin FCN katmanının başlangıç değerleri olarak atanır.
 - 5) CNN modelinin eğitim süreci başlatılır.
-

Sonuç olarak S-CNN modeli, aynı sınıfa ait örneklerin vektör toplamını kullanarak öğrenme sürecinin başlangıcında daha uygun bağlantı ağırlıklarıyla başlanmasını sağlamaktadır. Toplama işlemi ile elde edilen bu başlangıç ağırlıkları, ağırlık daha hızlı yakınsamasına ve daha yüksek doğruluk değerlerine ulaşmasına katkı sağlamaktadır.

2.2.4. Rastgele toplam tabanlı evrişimli sinir ağı (RS-CNN)

RS-CNN modeli, S-CNN gibi evrişim sonrası elde edilen vektörler içinden rastgele seçilen iki vektörden toplam vektörü oluşturup normalize edilmesiyle bağlantı ağırlıklarını üretir. S-CNN'den farklı olarak vektörlerin sınıflarına bakmaksızın tamamen rastgele seçilen vektörü direkt olarak aritmetik işleme alıp, bağlantı ağırlıklarını üretir. Bu model, daha önce önerilen D-CNN, C-CNN ve S-CNN yapılarına benzer biçimde çalışsa da vektör seçiminde sınıf benzerliklerine bakmaması diğer algoritmalardan farklı bir işlem oluşturmaktadır. RS-CNN modelinin akışı şeması Şekil 2.11'de şema olarak sunulmuştur.



Şekil 2.11 RS-CNN akış şeması.

RS-CNN modelinin işlem adımları Şekil 2.11’de verilen akış şemasında özetlenmiştir. Akış şemasında diğer kısıtlı modellerden farklı olarak seçilen sınıflarda aynı ya da farklı olması için ek bir döngüye ihtiyaç duymamasıdır. Seçilen vektörler sınıfına bakmaksızın hesaplamaya alınır. Ayrıca hesaplama adımı S-CNN de olduğu gibi toplam vektöründen yararlanılmaktadır. RS-CNN modelinin akışı uygularken kullanılan sözde kodu Algoritma 4’te verilmiştir.

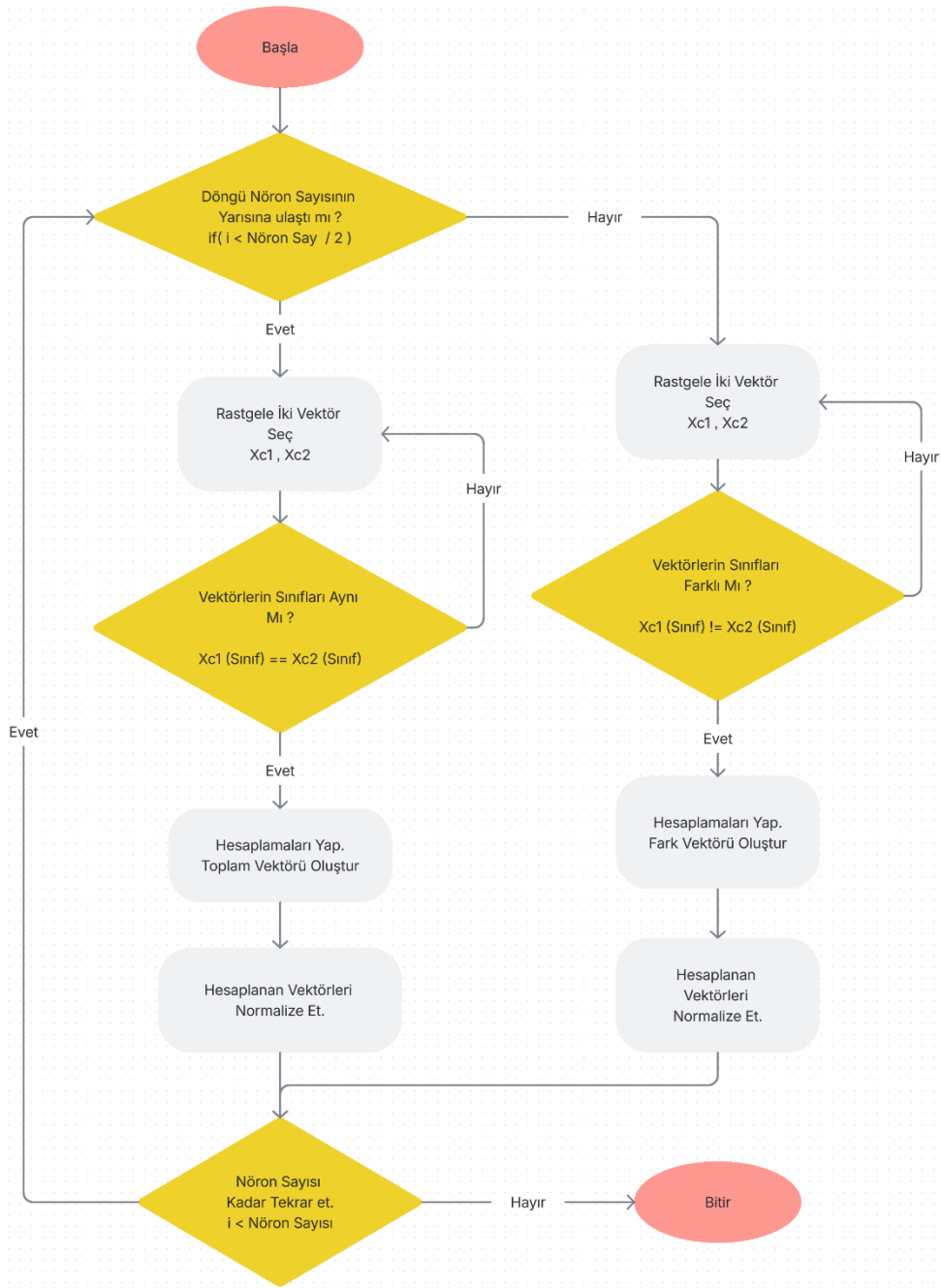
Algoritma 2.4: RS-CNN modelinin sözde kodu

- 1) Veri kümesi konvolüsyon işleminden geçirilir ve düzleştirme katmanından veri elde edilir.
 - 2) **for** $i = 1:N$ **do**
 - a) Rastgele iki sınıftan iki vektör X_{c1} ve X_{c2} seçilir
 - b) Toplam vektörü $X_{c2} + X_{c1}$ oluşturulur
 - c) Toplam vektörü normalleştirilerek bulunur: $W_i = \frac{2(X_{c2} - X_{c1})}{\|X_{c2} + X_{c1}\|}$
 - d) Elde edilen w_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı elde edilir.
 - 3) **end for**
 - 4) Elde edilen W matrisi CNN modelinin FCN katmanının başlangıç değerleri olarak atanır.
 - 5) CNN modelinin eğitim süreci başlatılır.
-

Sonuç olarak RS-CNN modeli, hesaplama adımı olarak S-CNN'e benzese de vektör seçimi olarak diğer önerilen modellerden ayrılan bir yapısı vardır. Bu farklılık ile beraber CNN'de istenilen sonuca yakınsamak için alternatif bir yaklaşım sunmaktadır.

2.2.5. Kısıtlı karma evrişimli sinir ağı (M-CNN)

M-CNN modeli, kısıtlı CNN modellerinin arasına diğerlerinden daha karmaşık bir yapıya sahiptir. Diğer kısıtlı modellerde nöron sayısı kadar bir döngü oluşturulup tek bir hesaplama yaklaşımı ile ağırlıklar üretilmişti. Bu yaklaşımda ise döngü ikiye ayrılmıştır. Döngünün yarısına kadar aynı sınıflardan seçilmiş iki vektörün toplamına dayanan bir yaklaşım, döngünün ikinci yarısında ise farklı sınıflardan seçilmiş iki vektörün farkının hesaplanmasına dayanan bir yaklaşım tercih edilmiştir. Bu oluşturulan karma model nöron sayısı kadar tekrar edilip istenilen ağırlık matrisi üretilmektedir. M-CNN bu yaklaşımıyla iki farklı yapıyı birleştiren bir algoritmadır. M-CNN'nin akış şeması Şekil 2.12'de verilmiştir.



Şekil 2.12 M-CNN ağındaki tam bağlı katman ağırlıklarının belirleme işlemleri akış şeması.

Şekil 2.12’de M-CNN modelinin ağırlık üretici bölümünün akışı sunulmuştur. Akış şemasına dikkat edildiğinde döngü ikiye ayrılmıştır. Döngüdeki sayaç nöron sayısının yarısına ulaşıncaya kadar işletilmektedir. Sayaç nöron sayısının yarısını geçtikten sonra o bölüm ise akış şemasında ilk olarak sağa dallanıp hesaplamaların yapıldığı bölümdür. Bu M-CNN akışını açıklayan sözde kod Algoritma 5’te verilmiştir.

Algoritma 2.5: M-CNN modelinin sözde kodu

- 1) Veri kümesi evrişim işleminden geçirilir ve düzleştirme katmanından veri elde edilir.
 - 2) **for** $i = 1:N$ **do**
 - a) **if** $i < N/2 + 1$ **then**
 - i) Rastgele olarak farklı iki sınıftan iki vektör X_{c1} ve X_{c2} seçilir
 - ii) Fark vektörü $X_{c2} - X_{c1}$ oluşturulur
 - iii) Fark vektörü normalleştirilerek bulunur: $W_i = \frac{2(X_{c2} - X_{c1})}{\|X_{c2} - X_{c1}\|}$
 - iv) Elde edilen w_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı elde edilir
 - b) **Else**
 - i) Rastgele iki aynı sınıfa ait iki veri noktası X_{c1} ve X_{c2} seçilir
 - ii) Toplam vektörü $X_{c2} + X_{c1}$ oluşturulur
 - iii) Toplam vektörü normalleştirilerek bulunur: $W_i = \frac{2(X_{c2} + X_{c1})}{\|X_{c2} + X_{c1}\|}$
 - iv) Elde edilen w_i kullanılarak $W_{i \times L}$ matrisinin i 'inci satırı elde edilir
 - c) **end if**
 - 3) **end for**
 - 4) Elde edilen W matrisi CNN modelinin FCN katmanının başlangıç değerleri olarak atanır.
 - 5) CNN modelinin eğitim süreci başlatılır.
-

Sonuç olarak M-CNN modeli, farklı ve rastgele sınıflar üzerinde gerçekleştirilen çıkarma ve toplama işlemlerini tek bir yapı içerisinde birleştirerek önerilen kısıtlı CNN modellerinin tek karma yaklaşımlı olanıdır. Önerilen bu model iki farklı yaklaşım arasında denge kurmayı sağlamaya çalışmaktadır.

2.3. Sezgisel Optimizasyon Algoritmaları

Sezgisel optimizasyon algoritmaları, karmaşık ve çözümü zor olan problemlerde, optimal çözüme ulaşmak için uygun çözümler bulmayı amaçlayan yöntemlerdir (Gandomi vd. 2013). Bu algoritmalar direkt olarak çözüme gitmek yerine genellikle deneyim veya sezgiye dayalı olarak problemi çözmek için uygun bir optimum çözüm arayan yöntemlerdir (Karaboğa 2004). Sezgisel optimizasyon algoritmalarının temel amacı global optimum çözüme, mümkün olduğunca yakın sonuçlar elde edebilmek ve bunu hesaplama maliyetini makul düzeyde tutarak gerçekleştirmektir. Bu temel amacı sağlayan literatürde birçok farklı algoritma geliştirilmiştir (Abdel-Basset vd. 2018; Hussain vd. 2019). Bu yöntemlerin bazıları hayvanların avlanma

stratejileri gibi doğadan, yaşamdan ya da optimizasyonu sağlamak için matematiksel yaklaşımlardan faydalanılarak geliştirilmiştir. Çözümüne giden optimizasyon yaklaşımları sürü tabanlı ve sürü tabanlı olmayan algoritmalar olarak ikiye ayrılabilir (Tunç vd. 2022). Bu çalışma kapsamında ilk olarak sürü tabanlı olmayan Tek Aday Optimizasyon Algoritması (SCO) kullanılarak bir model geliştirilmiştir. Çalışmanın devamında ise sürü tabanlı Yapay Arı Koloni Optimizasyon Algoritması (ABC) kullanılarak ayrı bir model geliştirilmiştir. Bu bölümde kullanılan SCO ve ABC algoritmalarının çalışma ilkeleri ve matematiksel temelleri anlatılmıştır.

2.3.1. Tek aday optimizasyon algoritması (SCO)

SCO algoritması T.M. Shami ve arkadaşları tarafından 2022 yılında geliştirilmiştir (Shami vd. 2024). Literatürde sıklıkla kullanılan algoritmalara bakıldığında genelde sürü tabanlı algoritmalar öne çıkmaktadır. Literatürdeki popüler yaklaşımın aksine SCO tek aday tabanlı bir optimizasyon algoritmasıdır. SCO algoritması tüm optimizasyon sürecinin yalnızca bir aday çözüm üzerinden optimize etmeye çalışır. Keşif ve yerel arama olarak iki aşamalı bir strateji ile güncelleme yöntemlerini çalıştırarak çözüme ulaşmaya çalışır (Balci vd. 2025).

SCO algoritması tek bir aday çözüm üzerinden başlatılmaktadır. Başlangıçtaki aday çözüm rastgele bir şekilde problemin doğal kısıtlar içerisinde başlatılabilmesi için Eşitlik D.8'e göre oluşturulur. Eşitlik D.8'de lb_i çözümün minimum alacağı değeri, ub_i aday çözümün alabileceği maksimum değeri, X_i aday çözümü, Rnd ise 0-1 aralığında rastgele seçilen bir değeri ifade etmektedir.

$$X_i = lb_i + Rnd (ub_i - lb_i) \quad (D.8)$$

SCO keşif ve yerel arama fazlarının ayrımını iterasyon sayısına göre takip eder. İki fazı birbirinden ayırmak için α eşik değerini kullanır. İterasyon α 'ya gelene kadar birinci faz olan keşif fazını çalıştırır, iterasyon sayısı α değerini aştığında ise yerel arama fazı çalışır. Bu faz ayrımını belirleyen α değeri, maksimum iterasyonun üçe bölünmesiyle elde edilir. Algoritma toplam iterasyon miktarının ilk üçte birlik bölümünde keşif yaparken, son üçte ikilik bölümde ise yerel arama yapar (Yuzgec 2025).

SCO optimizasyon algoritmasının birinci fazı olan keşif fazında güncelleme işlemleri

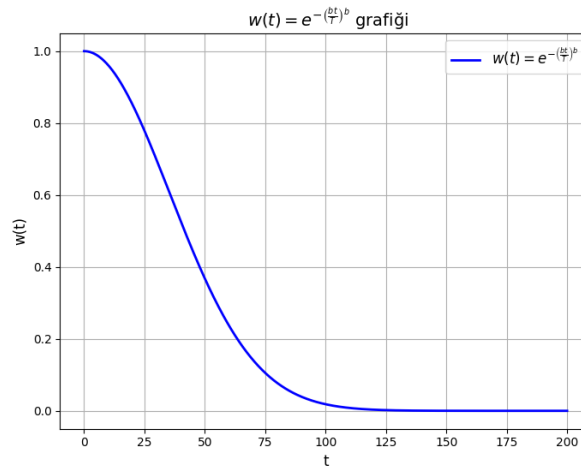
için tek bir güncelleme yöntemi kullanılmaktadır. Kullanılan güncelleme yöntemi Eşitlik D.9’da verilmiştir.

$$X_i = \begin{cases} gbest_i + w \times |gbest_i| & \text{Eğer } Rnd < 0,5 \\ gbest_i - w \times |gbest_i| & \end{cases} \quad (D.9)$$

Keşif fazında [0,1] aralığında rastgele bir değer seçilir. Seçilen değer 0,5’ten küçükse birinci duruma denk gelen toplam formülü uygulanırken, 0,5’ten büyükse ikinci durumdaki fark formülü uygulanır. Formüldeki $gbest_i$ en iyi adayın konumunu ifade etmektedir. Formülde kullanılan w iterasyona bağlı üretilen değeri ifade etmektedir.

$$w(t) = \exp^{-\left(\frac{bt}{T}\right)^b} \quad (D.10)$$

Eşitlik D.10’daki $w(t)$ hesaplamasında, b sabit bir değeri, t iterasyon sayısını ve T ise maksimum iterasyon sayısını ifade etmektedir. Eşitlik D.10’dan elde edilen w sabitinin 200 iterasyon için değişim grafiği Şekil 2.13’te verilmiştir.



Şekil 2.13 SCO güncelleme fonksiyonunda kullanılan $w(t)$ 'nin değişim grafiği.

SCO ikinci fazında iki farklı güncelleme fonksiyonu kullanır. İteratif olarak güncelleme olup olmadığını başarısız güncelleme sayacı isimli bir değişkende takip eder. Güncelleme olmadığında başarısız güncelleme sayacı değerini bir artırır ve bu değeri her iterasyonda başlangıçta belirlenen “ m ” eşliğini aşıp aşmadığı kontrol edilir. Eğer başarısız güncelleme sayacı eşiki aştıysa Eşitlik D.11’ye göre güncelleme yapılır ve sayaç sıfırlanır, eğer eşik

aşılmadıysa Eşitlik D.12'e göre güncelleme yapılır.

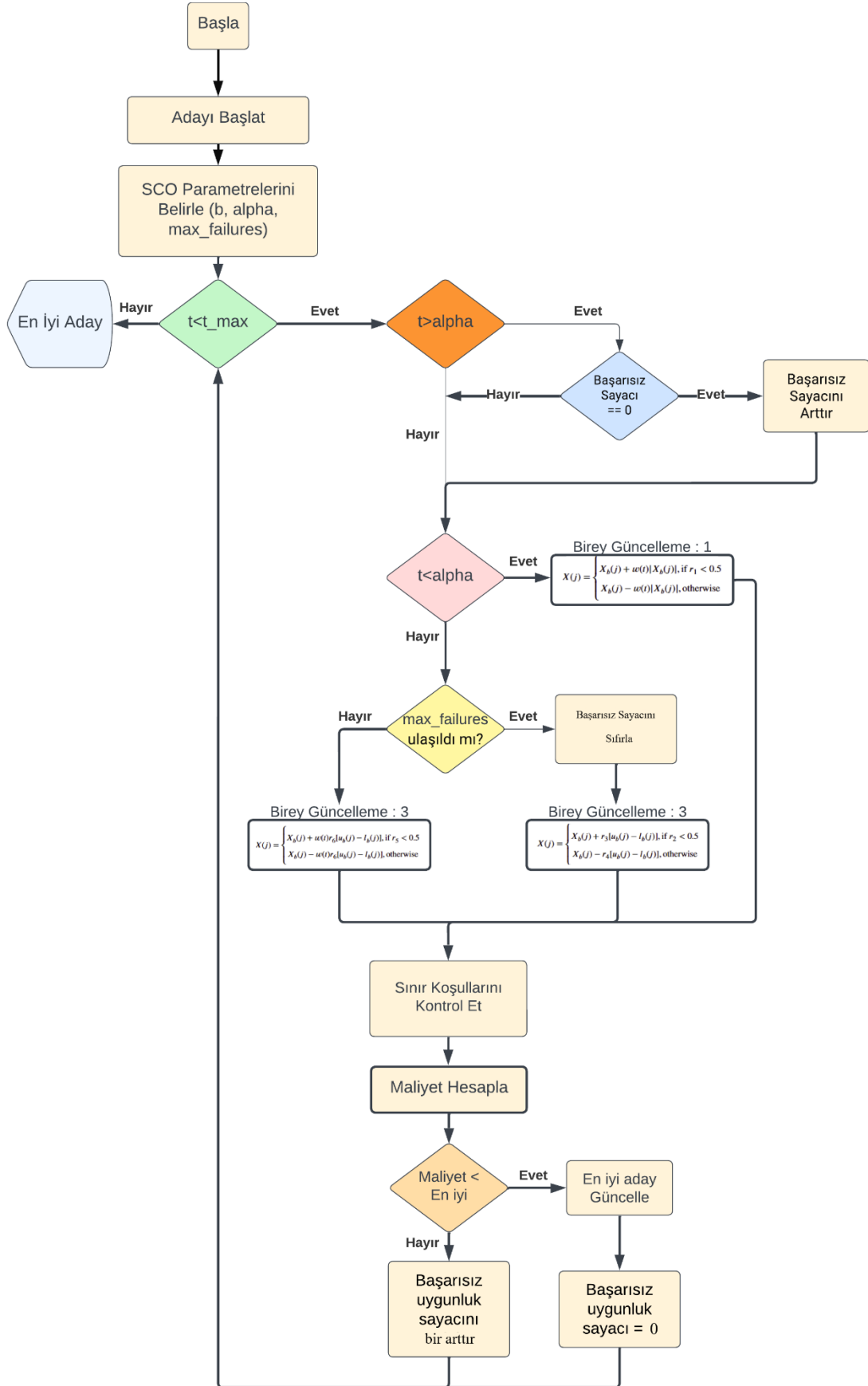
$$X_i = \begin{cases} gbest_i + Rnd \times w \times (ub_i - lb_i) & \text{Eğer } Rnd < 0,5 \\ gbest_i - Rnd \times w \times (ub_i - lb_i) & \end{cases} \quad (D.11)$$

$$X_i = \begin{cases} gbest_i + Rnd \times (ub_i - lb_i) & \text{Eğer } Rnd < 0,5 \\ gbest_i - Rnd \times (ub_i - lb_i) & \end{cases} \quad (D.12)$$

Eşitlik D.11 ve Eşitlik D.12 incelendiğinde, iki eşitlik arasında sadece w değeriyle çarpılma işlemi kaldırılarak elde edilmiştir. Yapılan güncellemeler sonucunda eğer aday, çözüm uzayının dışına çıkarsa Eşitlik D.13'e göre tekrar çözüm uzayının içinde kalacak şekilde güncelleme yapılır.

$$X_i = \begin{cases} gbest_i & \text{Eğer } X_i > ub_i \\ gbest_i & \text{Eğer } X_i < lb_i \end{cases} \quad (D.13)$$

Bu döngü başlangıçta belirlenen bir iterasyona ulaşıldığında algoritma sonlanır. Matematiksel hesaplamaları verilen SCO algoritmasının akış şeması Şekil 2.14'te sunulmuştur.



Şekil 2.14 SCO akış şeması.

Şekil 2.4'te sunulan akış şemasında gösterilen SCO süreci, Algoritma 2.6'da sözde kod olarak verilmiştir.

Algoritma 2.6: SCO algoritmasının sözde kodu

- 1) Alt ve üst sınırlar (lb , ub), boyut, uygunluk fonksiyonu (f), maksimum iterasyon sayısı (t_{max}), başarısız güncelleme sayacı (m), parametreler (α , b) tanımlanır.
 - 2) Başlangıç sürüsü oluşturulur
 - 3) Her bir boyut için Eşitlik D.5 ile hesaplanır.
 - 4) Başlangıç uygunluk değeri $BF = f(X)$ olarak atanır
 - 5) **for** $t = 1 \rightarrow t_{max}$ **do**
 - a) Ağırlık değeri Eşitlik D.11 ile hesaplanır.
 - b) Eğer $t > \alpha$ ve $FitImp = 0$ ise, m bir artırılır.
 - c) Rastgele katsayı $K = rand$ belirlenir.
 - d) **for** $j = 1 \rightarrow dim$ **do**
 - e) Eğer $t < \alpha$:
 - i) Eşitlik D.13 göre hesaplanır.
 - f) Aksi halde:
 - i) Eşitlik D.12'ye göre hesaplanır.
 - g) Sınır kontrolü yapılır:
 - i) Eşitlik D.14'ye göre hesaplanır.
 - h) Yeni çözümün uygunluğu hesaplanır: Eğer $f(X) < BF$ ise $BF = f(X)$,
Küresel en iyi çözüm $gbest = X_b$ olarak belirlenir.
 - 6) Döngü tamamlandığında en iyi uygunluk değeri ve çözüm olarak birey döndürülür.
-

2.3.2. Yapay arı koloni algoritması (ABC)

ABC algoritması, 2005 yılında Derviş Karaboğa tarafından geliştirilmiş doğadan esinlenen bir optimizasyon algoritmasıdır (Karaboga 2005). Bal arılarının doğal besin arama davranışını taklit eden bir matematiksel model ile geliştirilmiştir (Karaboga ve Basturk 2007). Çalışmada kullanılan SCO algoritmasından farklı olarak sürü tabanlı bir optimizasyon algoritmasıdır. Optimizasyon sürecinde işçi arılar, gözcü arılar ve kaşif arılar olmak üzere üç farklı arı modellenmiştir.

- İşçi arılar mevcut bir nektar kaynağına atanmış arıları temsil etmektedir.

Kaynağın kalitesini değerlendirir ve güncelleme işlemi yapar.

- Gözcü arılar ise işçi arıların bilgilerini gözlemlerler ve daha iyi kaynaklara yönelme eğilimindedirler.
- Kaşif arılar ise belirli bir iterasyon süresinde iyileşmeyen kaynakları terk edip rastgele bir kaynak ararlar (Karaboga 2005).

İlk olarak işçi arı fazı çalıştırılır bu fazda arıların her biri için Eşitlik D.14'e göre güncelleme yapılır. Eşitlikte ϕ_{ij} simgesi [0-1] aralığında bir rastgele seçilen bir değeri ifade eder. V_{ij} yeni üretilen bireyi, ifade etmektedir. X_{ij} mevcut arıyı, X_{kj} ise seçilen komşu arıyı ifade etmektedir.

$$V_{ij} = X_{ij} + \phi_{ij} (X_{ij} - X_{kj}) \quad (D.14)$$

Eşitlik D.14'teki hesaplamada üretilen arıların uygunluk değeri ile mevcut arının uygunluk değeri karşılaştırılır. Eğer yeni üretilen arı daha iyi uygunluk değerine sahip ise o arı ile devam edilir. İşçi arı fazından sonra gözcü arı fazına geçilir.

Gözcü arı fazında rulet tekerleği, turnuva gibi bir yöntem ile başlangıçta belirlenen sayıda rastgele arılar seçilir ve bu arılar için tekrar Eşitlik D.14'teki işleme tabi tutulur. Burada tekrar uygunluk değerleri kontrol edilir ve eğer uygun ise güncelleme yapılır.

İşçi ve gözcü fazlarında iteratif bir şekilde hesaplanırken her iterasyonda güncellememe sayıları tutulur. Belirlenen limitin üzerinde güncellenmeyen arılar için keşif fazı çalıştırılır. Bu fazda üretilen arılar mevcut pozisyonundan ve herhangi bir komşudan bağımsız olarak çözüm uzayında rastgele bir noktaya gönderilerek Eşitlik D.15'teki formül ile çözüm iyileştirilmeye çalışılır.

$$X_{ij} = X_{min j} + Rand(0,1) * (X_{max j} - X_{min j}) \quad (D.15)$$

Eşitlik D.15'de verilen matematiksel formül ile kaşif arı güncellemesi gerçekleştirilir. Bu faz çözüme giderken tıkanan noktalarda yeni çözüm noktaları bulmayı hedefler. Bu işlem yerel minimumlara takılmamak için etkili bir çözüm oluşturmaktadır (Karaboga ve Akay 2011). ABC algoritmasının sözde kodu Algoritma 2.7'de verilmiştir.

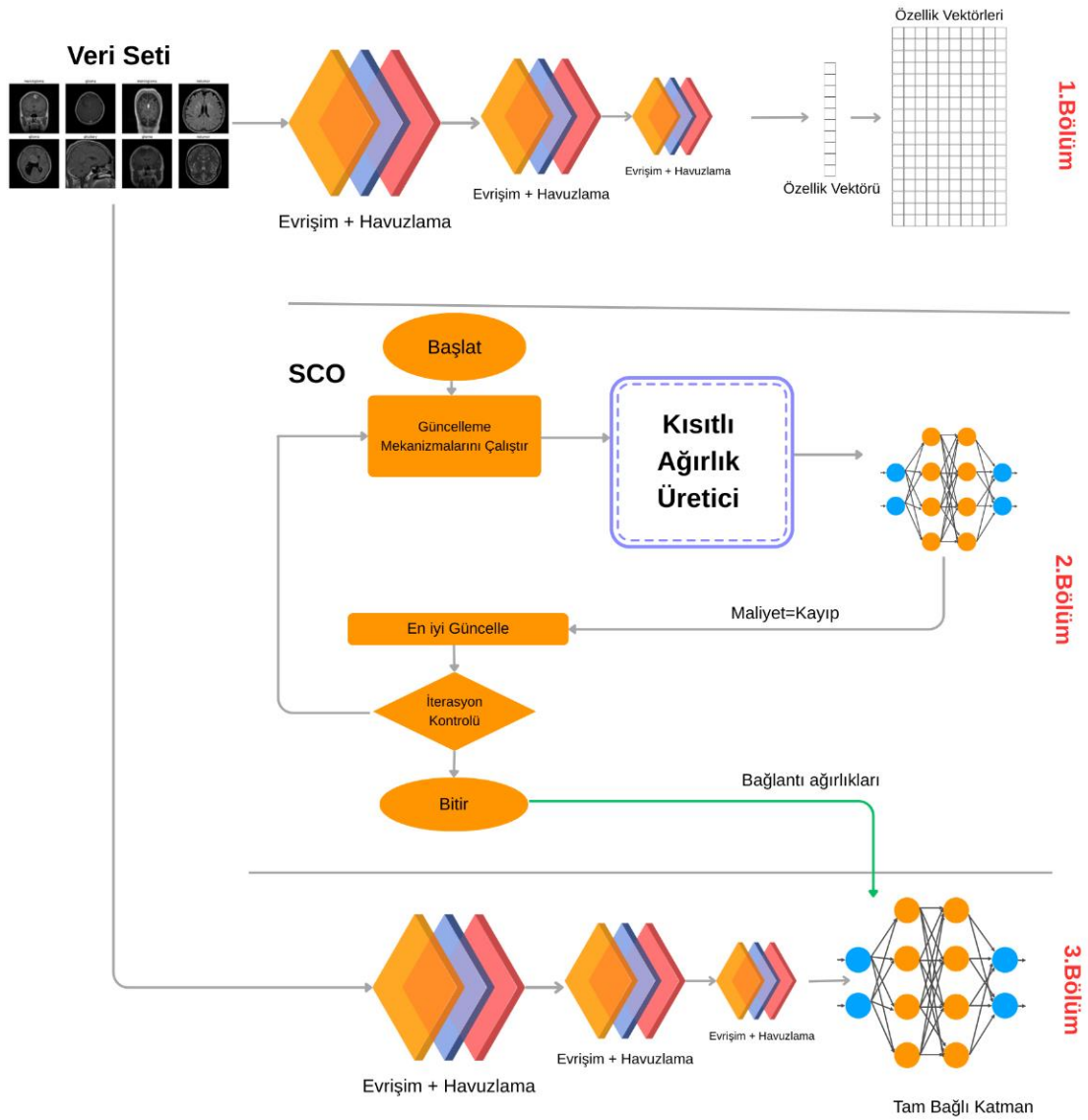
Algoritma 2.7: ABC algoritmasının sözde kodu

- 1) Başlangıç parametreleri belirlenir.
 - a) Arı sayısı (N), gözcü arı sayısı (ON), limit (güncellenmeme durumunda keşif fazını devreye alma limiti), güncellenmeme sayacı (trial), Çözüm uzayının sınırları, maksimum iterasyon (t_{max}) parametreleri belirlenir.
 - b) Başlangıç sürüsü oluşturulur.
 - c) Başlangıç sürüsü için uygunluk değerleri hesaplanır.
 - 2) for $t = 1 \rightarrow t_{max}$
 - a. İşçi arı fazı
 - (1) Yeni arılar (V_{ij}) üretilir. Eşitlik D.14'teki uygulanır.
 - (2) Uygunluk hesaplanır. $F(V_{ij})$
 - (3) Eğer $F(V_{ij}) < F(X_{ij})$ ise; $X_{ij} = V_{ij}$, değilse; trial = trial+1
 - b. Gözcü arı fazı
 - (1) Belirlenen sayıda gözcü arı sürüsünden rastgele seçilir.
 - (2) Yeni arılar (V_{ij}) üretilir. Eşitlik D.14'teki uygulanır.
 - (3) Uygunluk hesaplanır. $F(V_{ij})$
 - (4) Eğer $F(V_{ij}) < F(X_{ij})$ ise; $X_{ij} = V_{ij}$, değilse; trial = trial+1
 - c. Kaşif arı fazı
 - (1) Trial > Limit ise;
 - (a) Yeni arılar (V_{ij}) üretilir. Eşitlik D.15'teki uygulanır.
 - (b) Uygunluk hesaplanır. $F(V_{ij})$
 - (c) Eğer $F(V_{ij}) < F(X_{ij})$ ise; $X_{ij} = V_{ij}$
 - (d) Trial=0
 - d. En iyi çözüm güncelleme
 - (1) Eğer $f(X) < BF$ ise $BF = f(X)$, gbest = X
 - 3) Sonuç: gbest
-

2.4. OptiConNet: Tek Aday Sınıf Seçimi ile Optimize Edilmiş Karma Kısıtlı CNN Modeli

OptiConNet, bu tez çalışması kapsamında önerilen yeni bir model olup, temelinde SCO

algoritması ile optimize edilen kısıtlı bir CNN modelidir. Model, çalışmanın önceki bölümlerinde önerilen D-CNN ve C-CNN kısıtlı modellerinden esinlenilerek fark vektörü üzerinden hesaplama yapan, kısıtlı ağırlık üretici katmanının vektör seçiminin sezgisel algoritmalar ile yapıldığı karma bir yapıdır. Kısıtlı CNN mimarilerinde vektörlerin seçimi oldukça önemlidir. Modellerde vektör seçilirken aynı sınıf ya da farklı sınıflar olması uygulanan modele göre değişiklik göstermektedir. Hesaplama sürecinde en faydalı vektörün belirlenmesinin, sonuçlar üzerinde olumlu bir etki yaratacağı öngörülmüştür. Bu fikirden yola çıkarak vektör seçiminin yapılmasında tek aday optimizasyon algoritması (SCO) kullanılarak en uygun sınıfların seçimi sağlanmıştır. Oluşturulan modelin mimarisi Şekil 2.15'te 3 bölümde gösterilmiştir.



Şekil 2.15 OptiConNet modelinin mimarisi.

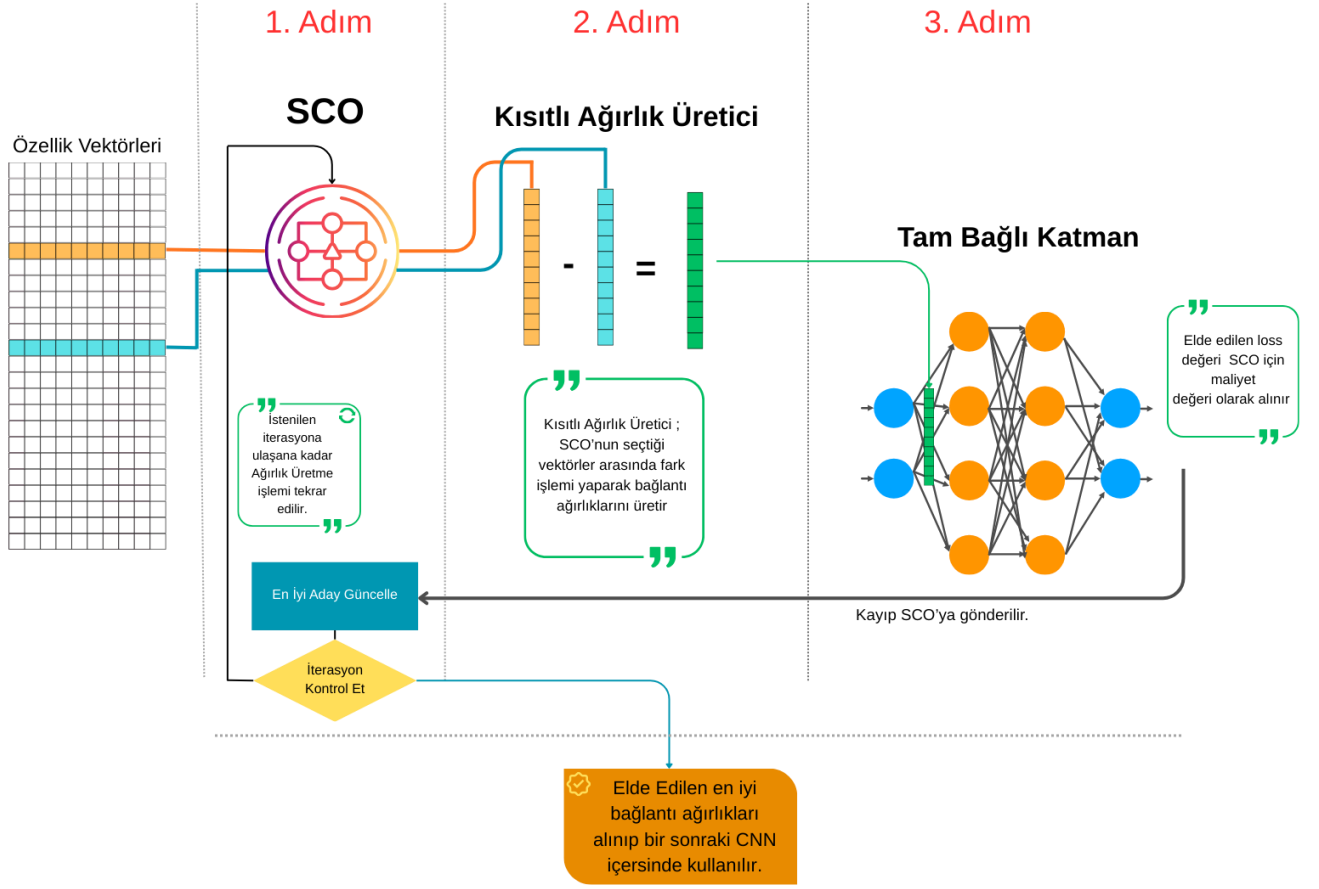
Önerilen OptiConNet modeli üç temel adımdan oluşmaktadır. İlk adımda, CNN modelinin özellik çıkarma süreçlerini gerçekleştiren ve çıktı olarak bir özellik vektörü üreten katman çalıştırılmaktadır. İkinci adımda SCO algoritması kullanılarak kısıtlı CNN hesaplaması uygulanmaktadır. Üçüncü adım ise, ikinci adımda üretilen ağırlıkları başlangıç noktası olarak kullanan klasik CNN eğitim yapısından oluşmaktadır.

Birinci bölümde, klasik CNN modelinin sadece özellik çıkarım katmanını kullanılmaktadır. Bu bölümde giriş verisi, Eşitlik D.1’de verilen matematiksel işlemler ile ardışık evrişim ve havuzlama işlemlerinden geçirilerek yüksek boyutlu bir özellik vektörüne dönüştürülür. Girdi görsellerinin boyutlarının sabit olması ve uygulanan işlemler sonucunda her örnek için aynı boyutta çıktılar üretilmesi sayesinde, bu özellik vektörleri düzleştirme katmanına aktarılmadan önce her bir örnek için sabit uzunlukta olacak şekilde yapılandırılır. Bu vektörler, ağırlık daha sonraki aşamalarında kullanılmak üzere bellekte saklanır. İkinci bölümde gerçekleştirilen hesaplamalar, bu bölümde elde edilen özellik vektörleri üzerinden yürütülmektedir. Ayrıca, üçüncü bölümde yer alan evrişim katmanları ile bu bölümdeki evrişim katmanları aynı parametre yapılandırmalarını paylaşmakta olup, böylece her iki bölümde üretilen özellik vektörlerinin tutarlı ve birbirleriyle uyumlu olması garanti altına alınmaktadır. Sonuç olarak, bu bölüm kısıtlı ağırlık üreticisinin doğru biçimde çalışabilmesi için gerekli olan alt yapıyı oluşturmakta ve üçüncü bölümde gerçekleştirilecek klasik CNN eğitiminin başlangıç noktası olarak işlev görmektedir.

İkinci bölüm, önerilen OptiConNet modelinin temel yeniliğini oluşturan ve CNN yöntemine katkı sağlayan kısımdır. Kısıtlı CNN modellerinde kısıtlı ağırlık üretici hesaplamada kullanacağı vektörleri rastgele seçerken, OptiConNet modelinde bu seçim SCO algoritması ile optimize edilmiştir. Böylelikle model daha uygun vektörler ile ağırlık üretme işlemini gerçekleştirmesi sağlanır. SCO algoritmasıyla birleştirilen kısıtlı ağırlık üretici üç ana adımda gerçekleştirilir:

- Birinci adım, SCO’nun bulunduğu vektör seçiminin optimize edildiği kısımdır.
- İkinci adım, SCO’nun seçtiği vektörler ile ağırlık üreticinin yeni ağırlıkları ürettiği kısımdır.
- Üçüncü adım, yeni üretilen ağırlıklarının uygunluğunun hesaplandığı içerisinde tam bağlı katman bulunan ve sadece ileri hesap yapıp kayıp maliyetinin hesaplandığı kısımdır.

Üç adımda gerçekleştirilen ikinci bölümde yapılan işlemlerin, işlem adımlarını anlatan görsel Şekil 2.16’da verilmiştir.



Şekil 2.16 OptiConNet mimarisinde ağırlık üretme sürecinin işlem adımları

Şekil 2.16’da verilen görselde ikinci bölümde yapılan ağırlık üretiminde nasıl bir yol izlendiği görülmektedir. Optimize edilmiş kısıtlı ağırlık üretilme süreci iç içe üç ana adımla gerçekleştirilmektedir.

Birinci adımda SCO algoritmasının ana görevi kısıtlı ağırlık üretici, için en uygun vektörleri seçmektir. SCO’da birey içerisinde ilgili vektörlerin indisini tutmaktadır. SCO algoritması birinci iterasyonda Eşitlik D.9’da verilen matematiksel yöntem ile aday birey vektör seçimini gerçekleştirir. Seçilen vektörler ikinci adımdaki kısıtlı ağırlık üreticiye gönderilir. İkinci ve üçüncü adımlar işletildikten sonra birinci adım tekrar devreye girer. Üçüncü adımda gelen kayıp değerine göre SCO’nun güncelleme yöntemlerini eğer keşif fazındaysa Eşitlik D.9’a göre, yerel arama fazındaysa güncellenmeme eşiğine bakarak Eşitlik D.11 veya Eşitlik D.12 ile güncelleme işlemini gerçekleştirir. SCO algoritması başlangıçta belirlenen iterasyon kadar kısıtlı ağırlık üretici için vektör seçimini gerçekleştirir ve üçüncü adımdan kendine gelen

kayıp değerine bakarak en uygun ağırlıkların olduğu vektör seçimini tutar.

İkinci adımda kısıtlı ağırlık üretici çalışır. OptiConNet modelinde kısıtlı ağırlık üretici vektör seçimini kendisi gerçekleştirmez, birinci adımda SCO algoritmasının seçtiği uygun vektörleri kullanarak hesaplamayı gerçekleştirir. Hesaplama yaklaşımı olarak ta kısıtlı CNN'lerden başarılı sonuçlar elde eden C-CNN ve D-CNN modellerinin fark vektörü oluşturma yaklaşımıyla hesaplamayı gerçekleştirir. Hesaplanan yeni ağırlıkların uygunluk değerini hesaplayabilmek için ağırlıklar üçüncü adıma gönderilir.

Üçüncü adımda CNN modelinin yalnızca tam bağlı katmanı işletilmektedir. Bu aşamada, ikinci adımda üretilen yeni ağırlıklar katmanın başlangıç ağırlıkları olarak atanır ve Eşitlik D.3 ve D.4'te tanımlanan ileri hesaplama süreci bir tur (epok) boyunca uygulanır. Epok sonunda elde edilen kayıp değeri, ilgili ağırlıkların uygunluğunu temsil eder. Hesaplanan bu kayıp değeri yeniden birinci adıma iletilerek SCO algoritmasının güncelleme süreci devam ettirilir ve iterasyon döngüsü sürdürülür.

Modelin ikinci bölümünde işletilen bu üç adım sonucunda son bölümde uygulanacak olan CNN için bütün hazırlıklar tamamlanmış olur. Burada elde edilen en uygun ağırlıklar CNN'de kullanılmak üzere üçüncü bölüme iletilir.

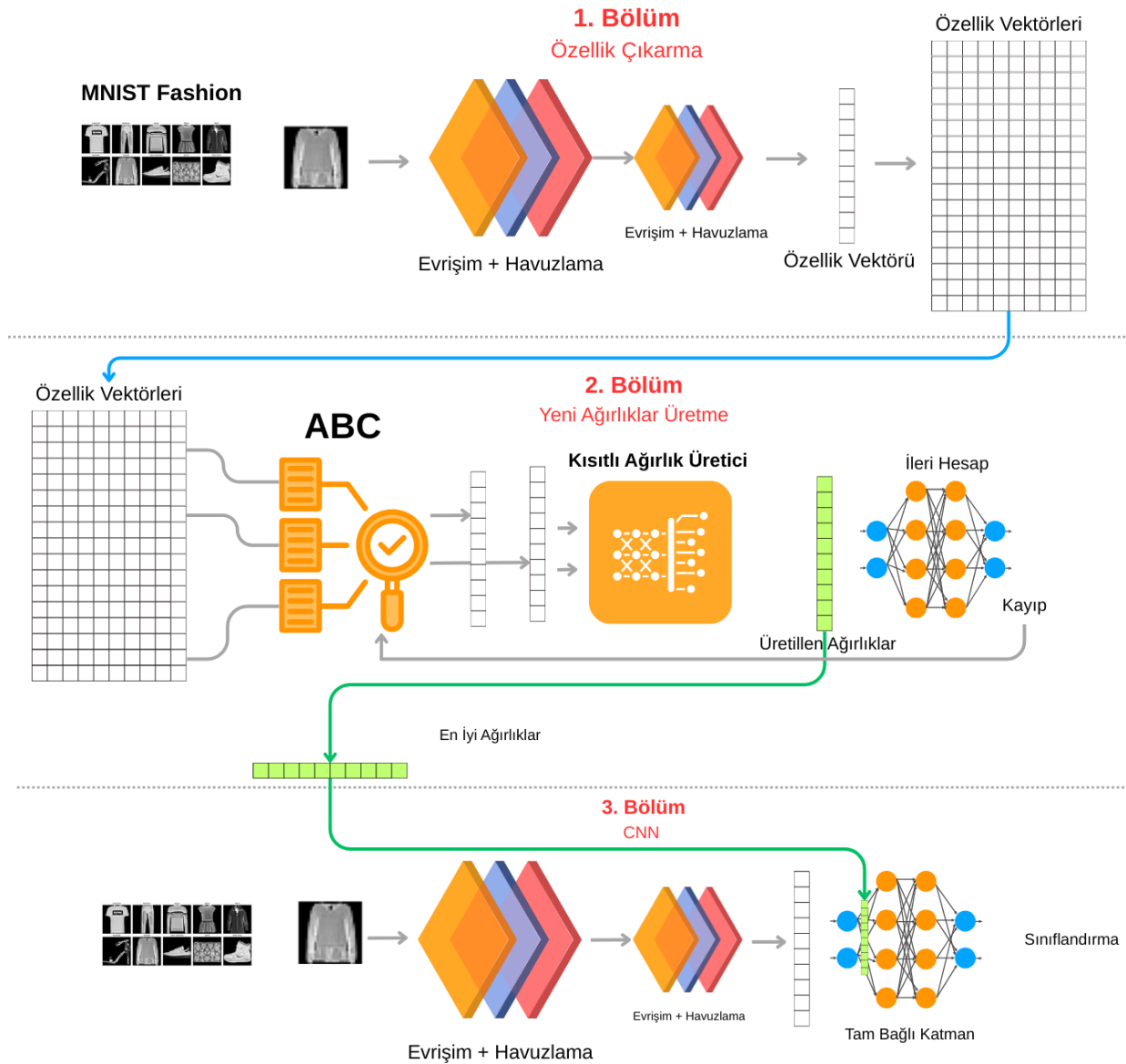
Modelin üçüncü bölümü, klasik CNN yapısının eğitim sürecinin gerçekleştirildiği aşamadır. Bu bölümde, kısıtlı ağırlık üretici tarafından optimize edilerek elde edilen bağlantı ağırlıkları, tam bağlantılı katmanların başlangıç parametreleri olarak kullanılmaktadır. Böylece model optimize edilmiş bir başlangıç noktasından öğrenme sürecine girer. Bu bölümde kullanılan klasik CNN modelinin parametreleri birinci bölümdeki parametreler ile aynı parametreler olacak şekilde model yapılandırılmıştır. Bunun sebebi ilk bölümde üretilen vektörler ile eş değer vektörlerin üretilmesini sağlamaktır. Eğitime başlanmadan önce kısıtlı ağırlık üretici tarafından üretilen en uygun ağırlıklar ağırlık başlangıç parametresi olarak atanıp eğitim başlatılır. Böylelikle model eğitime başlarken, çözüme daha yakın bir noktadan başlatılmış olur. Bu sayede daha az bir tur (epok) ile ağırlıkların istenilen seviye sonuçlar almasını sağlamaktadır. Bu bölümdeki eğitilen model ile sınıflandırma yapılmaktadır.

Sonuç olarak OptiConNet mimarisi, üç aşamalı bütünleşik yapısı sayesinde klasik CNN tabanlı modellerden farklı bir yaklaşım sergilemektedir. Birinci bölümde veri kümesi, evrimsel katmanlardan geçirilerek anlamlı özellik vektörleri çıkartılır, ikinci bölümde optimize edilmiş kısıtlı ağırlık üretici ile ağırlıklar üretilir, üçüncü bölümde ise üretilen

ağırlıklar ile eğitim gerçekleştirilir. İkinci bölüm, modelin en yenilikçi kısmını oluşturan kısıtlı ağırlık üretici yapıdır. Bu kısımda SCO tabanlı optimizasyon yaklaşımı kullanılarak, tam bağlantılı katmanlarda kullanılacak ağırlıkların başlangıç değerleri dinamik biçimde hesaplanmakta, böylece modelin eğitime üstün bir başlangıç sağlanmaktadır. Bu sayede OptiConNet mimarisi, klasik CNN modellerinin öğrenme başarımını artıran ve doğruluk oranlarını yükselten karma bir yaklaşım olarak değerlendirilebilir.

2.5. HyBee-CNN: Yapay Arı Kolonisi ile Optimizasyonu Gerçekleştirilen Kısıtlı CNN Modeli

HyBee-CNN modeli, tez çalışması kapsamında önerilen diğer modeller gibi CNN modelinin başlangıçtaki rastgele belirlenen parametrelerini optimize etmek için önerilmiş yeni bir modeldir. HyBee-CNN, Kısıtlı CNN modelleri içerisine ABC algoritması eklenerek oluşturulmuş karma bir modeldir. Tez çalışması kapsamında önerilen OptiConNet modeli ile benzer yapıda oluşturulmuştur. OptiConNet içerisinde vektör seçimi tek aday tabanlı SCO algoritması ile yapılırken, HyBee-CNN modelin de vektör seçimi sürü tabanlı ABC algoritması ile gerçekleştirilmiştir. Modelin oluşturulan çalışma yapısı Şekil 2.17'de sunulmuştur.



Şekil 2.17 HyBee-CNN modelinin mimarisi.

Şekil 2.17’de verilen HyBee-CNN modeli temel olarak üç ana bölümden oluşmaktadır. Birinci ve üçüncü bölüm OptiConNet modeliyle benzer şekilde yapılandırılmıştır. İkinci bölümde ise SCO yerine ABC algoritması kullanılmıştır.

Birinci bölümde CNN’nin sadece özellik çıkarım katmanları çalıştırılır ve Eşitlik D.1’de verilen matematiksel yöntem ile veri kümesinden özellik vektörleri elde edilir. Elde edilen bu özellik vektörleri ikinci bölümde kullanılmak üzere hafızaya alınır.

İkinci bölümde veri kaynağı olarak hafızaya alınan özellik vektörleri kullanılır. HyBee-CNN modelinin ağırlık üretme süreci, OptiConNet ile benzer akışla ilerleyen yapıda oluşturulmuştur. Oluşturulan modelde ağırlık üreticinin hesaplamada kullanacağı vektörleri

ABC algoritması ile belirlenir. Ağırlık üretilirken vektör farklarının hesaplanabilmesi için nöron sayısı kadar vektör ikilisine ihtiyaç duyulmaktadır. ABC Algoritmasında bulunan her bir arı, nöron sayısı kadar ikili vektör indisi tutmaktadır. Bu çift olarak tutulan vektör indisleri ağırlık üreticiye verilir. Ağırlık üretici indislere karşılık gelen vektörleri okur ve vektörler arasında fark işlemi yaparak ağırlıklar üretilir. Ağırlık üretici tarafından üretilen yeni ağırlıklar test için oluşturulan tam bağlı katmanda ileri hesap yapıp kayıp değeri elde edilir. Bu katmanda elde edilen kayıp değeri ABC algoritmasındaki bir arının maliyeti olarak ele alınır. Bu maliyet değerine göre ABC algoritması güncelleme olup olmayacağına karar verir. Eğer güncelleme yapılacaksa Eşitlik D.14'te verilen matematiksel işlem ile güncellemeleri gerçekleştirir. Arıların güncellenmeme sayacı belirlenen limiti aştığında güncelleme Eşitlik D.15'e göre yapılır. Bu işlem iteratif bir şekilde tekrar edilir. Başlangıçta belirlenen iterasyon sayısı kadar bu işlem tekrar edilip en uygun ağırlıklar CNN ağına kullanılması için üçüncü bölüme gönderilir.

Üçüncü bölümde üretilen ağırlıklar CNN'in tam bağlı katmanına atanır. Bu başlangıç parametreleri ile model eğitilir. Bu bölümde yapılan işlemler de OptiConNet ile benzer şekilde yürütülmektedir.

HyBee-CNN modelinin, önceki bölümde sunulan OptiConNet modelinden ayrıldığı temel nokta, ağırlık optimizasyon sürecinde SCO algoritması yerine ABC algoritmasının tercih edilmesidir. Bu tercih, iki optimizasyon yönteminin yapısal farklılıklarından kaynaklanmaktadır. ABC algoritması sürü tabanlı bir arama stratejisi benimserken, SCO algoritması tek aday çözüm üzerinden çalışan birey tabanlı bir yapıya sahiptir. Bununla birlikte, SCO algoritması üzerine literatürde yapılan çalışmaların sınırlı sayıda olması; ABC algoritmasının ise uzun yıllardır geniş bir uygulama alanına sahip, iyi bilinen ve güvenilir bir meta-sezgisel yöntem olarak kabul edilmesi, önerilen mimarinin daha yaygın kullanılan bir optimizasyon yaklaşımıyla bütünleştirilmesine imkân sağlamıştır.

2.6. Veri Setleri

Bu çalışma kapsamında önerilen kısıtlı CNN modelleri; MNIST, MNIST Fashion ve CIFAR-10 veri kümeleri üzerinde test edilmiştir. OptiConNet modeli ise MNIST Fashion ve Brain Tumor MRI veri kümeleri üzerinde test edilmiştir. HyBee-CNN modeli ise sadece MNIST Fashion veri kümesi üzerinde eğitilip test edilmiştir. Kullanılan veri kümelerinin temel özellikleri Tablo 2.2'de özet olarak verilmiştir.

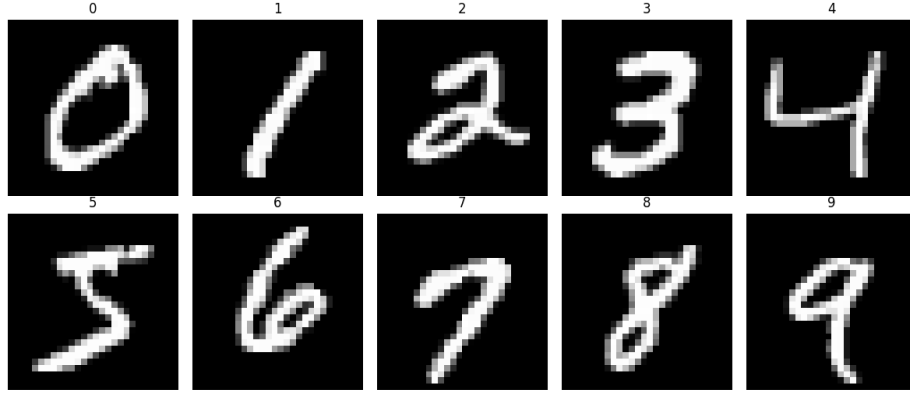
Tablo 2.2 Veri kümelerinin temel özellikleri.

Veri Kümesi	Boyut	Eğitim / Test	Veri Sayısı	Sınıf Sayısı
MNIST	28×28	60000 / 10000	70000	10
MNIST Fashion	28×28	60000 / 10000	70000	10
CIFAR-10	32×32	50000 / 10000	60000	10
Brain Tumor MRI	128×128	5712 / 1311	7023	4

Tablo 2.2 incelendiğinde MNIST, MNIST Fashion ve CIFAR -10 veri kümelerinin daha küçük boyutlu resimlerden oluştuğu Brain Tumor MRI veri kümesinin daha yüksek boyutlu resimlerden oluştuğu görülmektedir. Ayrıca MNIST, MNIST Fashion ve Brain Tumor MRI veri kümeleri gri seviyeli görseller içerirken CIFAR-10 RGB renkli görsellerden oluşmaktadır. MNIST, MNIST Fashion ve CIFAR -10 veri kümelerinde 10 farklı sınıf bulunurken, Brain Tumor MRI veri kümesinde 4 farklı sınıf bulunmaktadır. Çalışmada kullanılan bu veri kümelerinin sınıf dağılımları ve detaylı incelemeleri bu alt bölümün devamında ayrı başlıklar halinde sunulmuştur.

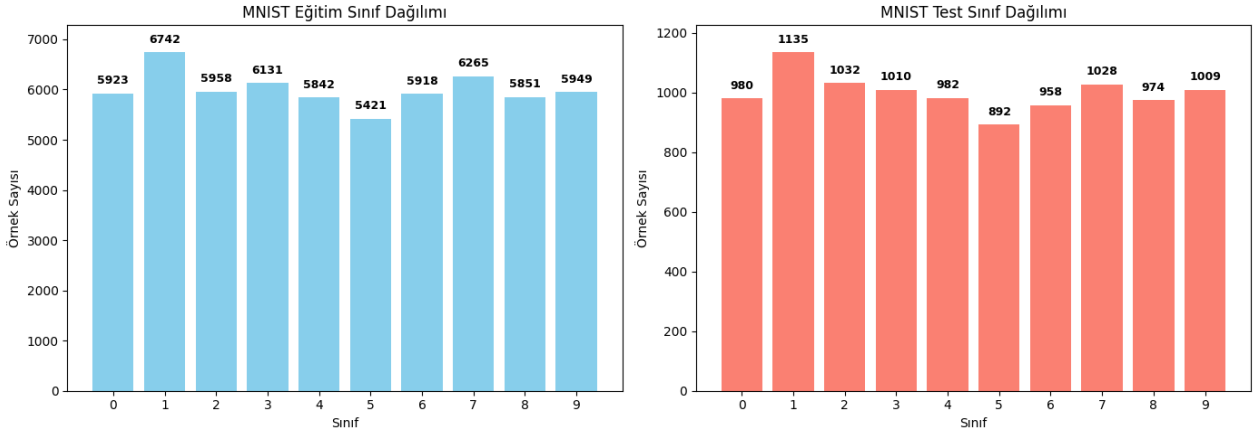
2.6.1. MNIST veri kümesi

MNIST veri kümesi el yazıyla yazılmış rakamları içeren 1994 yılında oluşturulmuş standart bir veri kümesidir (LeCun 1998b). Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından hazırlanmış olan “NIST Special Database 1” ve “NIST Special Database 3” veri kümelerinin birleştirilmesi ve yeniden düzenlenmesi sonucunda oluşturulmuştur. Oluşturulduğu tarihten günümüze kadar görüntü işleme ve derin öğrenme çalışmalarında sıklıkla kullanılan veri kümelerinden biri olmuştur (Grim ve Somol t.y.; Kadam vd. 2020). Literatürde çok sayıda kullanılması, farklı çalışmaların birbiriyle karşılaştırılması için standart hale gelmiştir. MNIST veri kümesinin örnek görselleri Şekil 2.18’de görülmektedir.



Şekil 2.18 MNIST veri kümesi örnek görselleri.

MNIST, içerisinde el yazısıyla oluşturulmuş toplam 70,000 görselden oluşan bir veri kümesidir. Görsellerin 60000 tanesi eğitim, 10000 tanesi ise test için ayrılmıştır. Tüm görseller 28×28 piksel çözünürlüğünde, gri tonlamalı ve normalize edilmiş biçimde hazırlanmıştır. 0-9 rakamları içeren bu veri kümesi toplam 10 tane sınıfa sahiptir. Şekil 2.19'da eğitim ve test sınıflarının dağılımları verilmiştir.



Şekil 2.19 MNIST veri kümesi eğitim ve test sınıf dağılım grafiği.

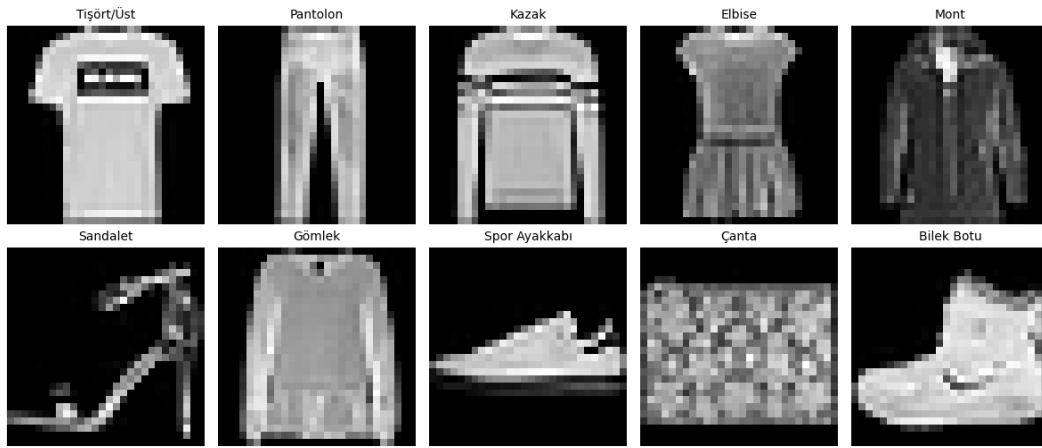
Şekil 2.19 incelendiğinde veri kümesi tamamen dengeli bir dağılıma sahip değildir. Eğitim veri kümesi için en çok veri sayısına sahip “1” sınıfı ortalamadan %12,37 fazlayken, en az veriye sahip “5” sınıfı ise ortalamadan %9,65 daha azdır. Test verilerinin dağılımı da eğitim verilerine benzer oradadır. Test tarafında aynı eğitimde olduğu gibi en fazla “1” sınıfına ait veri varken en az “5” sınıfına ait veri vardır. Veri kümesi dengesiz olsa da toplamdaki veri sayısı ve sınıf sayısı göz önüne alındığında sınıflar arasındaki dağılım farkları çok yüksek değildir.

Veri kümesindeki verilerin boyutlarının küçük olması, çok sayıda görsel içermesi ve literatürde sıklıkla kullanılması bu sayede farklı çalışmalar için karşılaştırma imkânı sunması,

veri kümesinin kullanılabilirliğini arttırmaktadır (Baldominos vd. 2019). Bu sebeple çalışmada MNIST veri kümesi tercih edilmiştir.

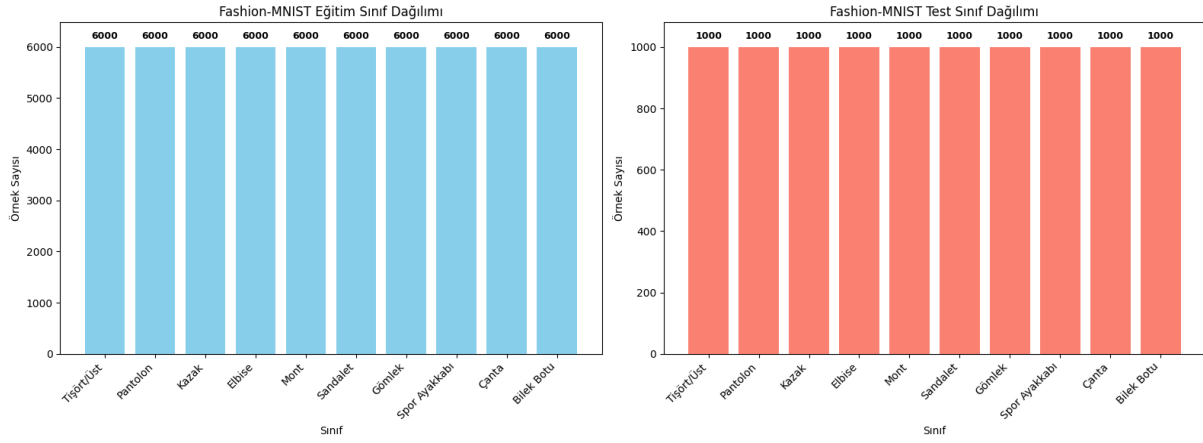
2.6.2. MNIST Fashion veri kümesi

MNIST Fashion veri kümesi 2017 yılında, MNIST veri kümesine alternatif olarak üretilmiş, içerisinde farklı moda ürünleri görselleri bulunduran bir veri kümesidir (Xiao vd. 2017). MNIST veri kümesinde el yazması sayıların sınıflandırılmasında geliştirilen algoritmaların çok yüksek doğruluk değerleri elde etmesi, modellerin başarımını değerlendirmekte sınırlı kalmıştır. Bu problemi giderme amacıyla, modelleri MNIST'e göre daha zorlayıcı bir veri kümesi olarak sunulmuştur. MNIST Fashion veri kümesinde görsellerin ayırt edilmesinin daha zor olması, modellerin gerçek dünyada karşılaşılabilecekleri problemleri daha güzel örnekleme için tasarlanmıştır (Bbouzidi vd. 2024). MNIST Fashion veri kümesinin örnek görselleri Şekil 2.20'de verilmiştir.



Şekil 2.20 MNIST Fashion veri kümesi örnek görselleri.

Şekil 2.20'de verildiği üzere MNIST Fashion veri kümesindeki görseller gri seviyeli olarak kaydedilmiştir. Görsellerin boyu 28×28 boyutunda tutulmaktadır. Veri kümesinde toplam 70.000 adet görsel bulunmaktadır. Bu verilerin 60.000 (%85,7) adeti eğitim için 10.000 (%14,3) adeti ise test için ayrılmıştır. MNIST Fashion veri kümesi toplamda 10 farklı sınıf bulunmaktadır. Eğitim veri kümesi ve test veri kümelerinin sınıf dağılımları Şekil 2.21'de verilmiştir.



Şekil 2.21 MNIST Fashion veri kümesi eğitim ve test sınıf dağılım grafiği.

Veri kümesindeki dağılımda her bir sınıftan eşit miktarda görsel veri içerdiği görülmektedir. Bu dağılım veri kümesinin dengeli olduğunu göstermektedir. Modellerin başarımları karşılaştırılırken dengeli veri kümeleri tercih edilirken, günlük hayat problemlerinde ise dengesiz dağılımlı veriler ile karşılaşılmaktadır.

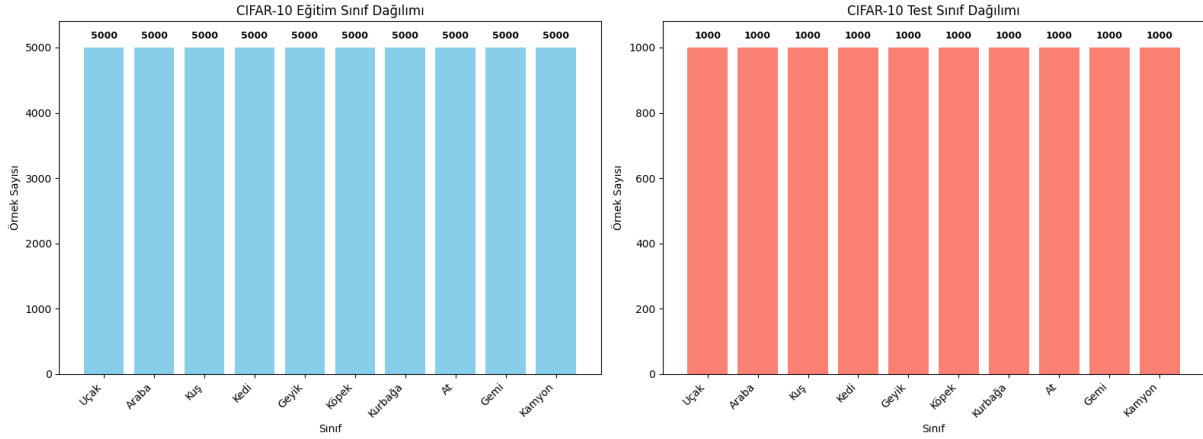
2.6.3. CIFAR-10 veri kümesi

CIFAR-10 veri kümesi, Alex Krizhevsky, Vinod Nair ve Geoffrey Hinton tarafından Toronto üniversitesinde 2008 yılında toplanıp etiketlenmiş ve 2009 yılında erişime açılmıştır (Krizhevsky ve Hinton 2009). Veri kümesi, özellikle nesne sınıflandırma ve görüntü tabanlı öğrenme algoritmalarının değerlendirilmesinde literatürde sıklıkla kullanılmaktadır (Doon vd. 2018; Obaid vd. 2020). Veri kümesi içerisinde günlük hayatta sıklıkla karşılaşılan 10 farklı nesne görsel olarak bulunmaktadır. Görüntülerin tamamı RGB renkli görsellerden oluşmaktadır. Veri kümesinde bulunan örnek görseller Şekil 2.22’de sunulmuştur.



Şekil 2.22 CIFAR-10 veri kümesi örnek görselleri.

Şekil 2.22’de örnek görselleri bulunan CIFAR-10 veri kümesinde toplam 60.000 adet veri bulunmaktadır. Bu görsellerin 50.000 tanesi eğitim, 10.000 tanesi ise test için ayrılmıştır. Resimlerin boyutları 32×32 pikseldir. Veri kümesinde toplam 10 farklı sınıfa ait veriler bulunmaktadır. Eğitim veri kümesi ve test veri kümelerinin sınıf dağılımları Şekil 2.23’de verilmiştir.



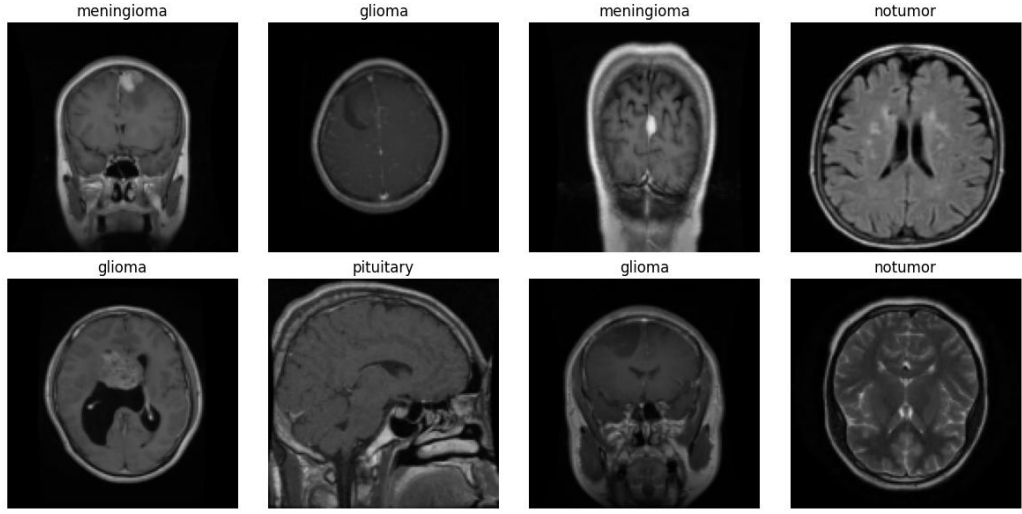
Şekil 2.23 CIFAR-10 veri kümesi eğitim ve test sınıf dağılım grafiği.

Şekil 2.23’deki grafikte veri kümesinin eğitim ve test veri kümelerinin eşit sayıda sınıflara dağıldığı dengeli bir dağılıma sahip olduğu görülmektedir. Veri kümesi düşük çözünürlüklü ama yüksek varyansa sahiptir. Bu sebeple literatürde sıklıkla tercih edilmektedir. Veri kümesi MNIST ve MNIST Fashion veri kümelerine göre model doğruluklarının daha düşük olduğu bir veri kümesidir. Çalışmada modelin daha zor veri kümesinde test edilmesi için tercih edilmiştir.

2.6.4. Brain Tumor MRI veri kümesi

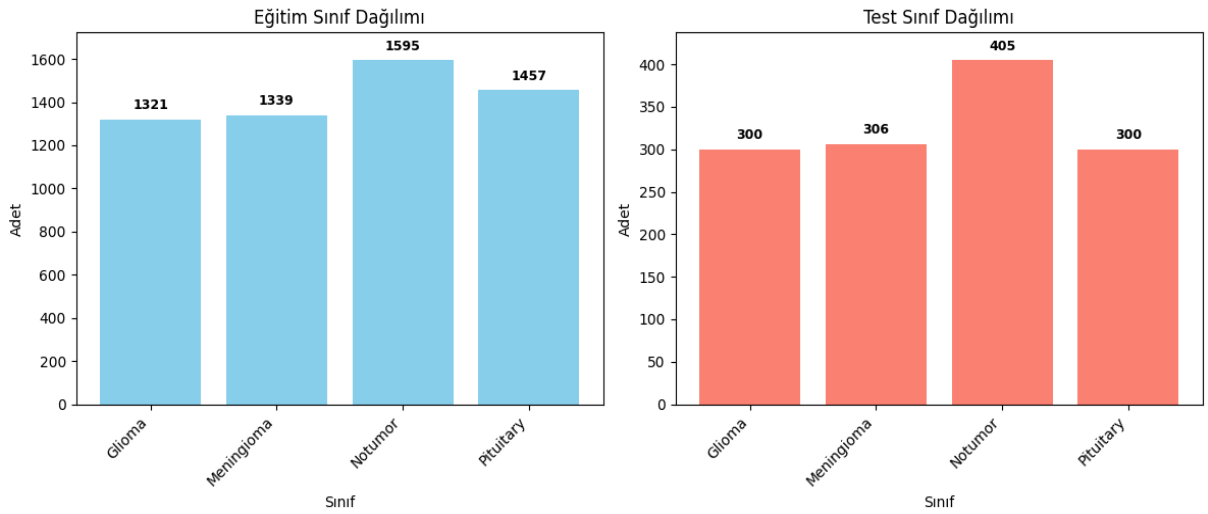
Brain Tumor MRI veri kümesi, çağımızın büyük sağlık problemlerinden biri olan kanser hastalığı için beyin bölgesinde Manyetik Rezonans Görüntüleme (MRI) yöntemiyle elde edilen görüntüleri içermektedir. Brain Tumor MRI veri kümesi, kanser tanısı, tespiti ve kanserli bölgenin yerinin belirlenmesi üzerine yapılacak yapay zekâ çalışmaları için önemli bir veri kaynağıdır. Bu veri kümesi daha önce oluşturulan Br35H, Brain Tumor Dataset, Brain Tumor Classification (MRI) veri kümelerinin birleşiminden oluşturulmuş bir veri kümesidir. Oluşturulan veri kümesi *glioma*, *meningioma*, *pituitary* ve *no tumor* olmak üzere üç farklı tümör türü hastalığı ve bir sağlıklı MRI görüntüsünden oluşan toplam 4 sınıfa sahiptir. Veri kümesinin

örnek görselleri Şekil 2.24’te verilmiştir.



Şekil 2.24 Brain Tumor MRI veri kümesi örnek görselleri.

Şekil 2.24’te verilen görsellerdeki gibi veri kümesinin tamamı gri seviyeli görüntülerden oluşmaktadır. Veri kümesindeki görüntülerin boyutları 128×128 pikseldir. Toplam dört sınıfa sahip olan veri kümesinde üç sınıf tümörlü olan görüntüleri içerirken sağlıklı görüntüleri içeren bir sınıf bulunmaktadır. Veri kümesinde toplam 6023 görsel bulunmaktadır. Bu görsellerin 5712 tanesi eğitim için, 1311 tanesi test işlemleri için ayrılmıştır. Bu dağılım ile veri kümesinin %81,3’ü eğitim, %18,7’si test için ayrılmış olmuştur. Veri kümesinde sınıfların dağılımı genel olarak birbirine yakın olsa da en çok “No tumor” sınıfına ait veriler bulunmaktadır. Eğitim veri kümesi ve test veri kümelerinin sınıf dağılımları Şekil 2.25’te verilmiştir.



Şekil 2.25 MNIST veri kümesi eğitim ve test sınıf dağılım grafiği.

Veri kümesindeki dağılımlar incelendiğinde hem eğitim verilerinde hem de test verilerin tümörlü sınıfların sayıları birbirine çok yakındır. Tümörsüz sınıfına ait veriler eğitim verilerin yaklaşık %20 oranında diğer sınıflardan fazladır, test verilerinde ise yaklaşık %30 daha fazladır.

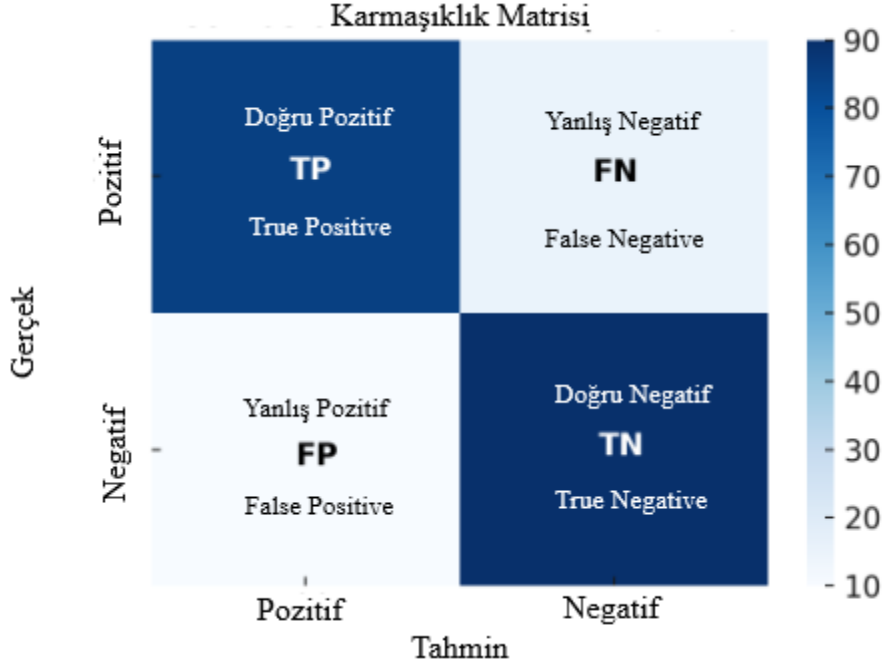
Çalışma kapsamında kullanılan diğer veri kümeleri kıyaslama için oluşturulan veri kümeleriydi, ancak Brain Tumor MRI veri kümesi önerilen OptiConNet modelini gerçek hayat problemlerindeki başarıyı gözlemlemek için tercih edilmiştir.

2.7. Başarım Değerlendirme Metrikleri

Makine öğrenmesi ve derin öğrenme yöntemlerinin değerlendirilmesinde kullanılan başarımların metrikleri, modellerin güçlü ve zayıf yönlerinin kapsamlı biçimde analiz edilmesine olanak sağlamaktadır. Özellikle sınıflandırma problemlerinde birden fazla metriğin birlikte kullanılması, yalnızca doğruluk değerine odaklanmak yerine modelin tüm yönleriyle bütüncül olarak değerlendirilmesine imkân tanımaktadır.

Bu çalışmada önerilen modellerin başarımları, doğruluk (accuracy), kesinlik (precision), duyarlılık (recall veya sensitivity), F1 Skoru, karmaşıklık matrisi ve AUC-ROC eğrisi gibi literatürde yaygın biçimde kullanılan metrikler üzerinden incelenmiştir (Madakam vd. 2022; Obi 2023). Deneysel sonuçlar ve tartışma bölümünde, literatürdeki çalışmalarla yapılan karşılaştırma tablolarında yalnızca doğruluk değerine yer verilmiş olsa da modelin kendi iç değerlendirmelerinde ve yüksek başarımlar gösterdiği tekrar deneylerinde tüm başarımların ayrıntılı biçimde kullanılmıştır.

Sınıflandırma problemlerinde kullanılan temel değerlendirme araçlarından biri karmaşıklık matrisidir (Sathyanarayanan ve Tantri 2024). Karmaşıklık matrisi, modelin her bir sınıfı ne ölçüde doğru tahmin ettiğini göstermenin yanı sıra, yanlış sınıflandırmaların hangi sınıflar arasında gerçekleştiğini de ayrıntılı biçimde incelememize olanak tanır. Özellikle ikiden fazla sınıftan oluşan veri kümelerinde, modelin bir sınıfı hangi diğer sınıflarla karıştırdığını açıkça gözlemlemek mümkündür (Fahmy Amin 2023). Matris oluşturulurken sınıf etiketleri hem satırlarda hem de sütunlarda yer alır. Matrisin köşegenindeki değerler modelin doğru sınıflandırdığı örnekleri, köşegen dışındaki değerler ise yanlış sınıflandırmaları temsil eder. Bu doğru ve yanlış sınıflandırmalar üzerinden Yanlış Pozitif (False Positive, FP), Yanlış Negatif (False Negative, FN), Doğru Pozitif (True Positive, TP) ve Doğru Negatif (True Negative, TN) başarımların ölçütleri tanımlanmaktadır. Bu ölçütler, Şekil 2.26'da verilen örnek karmaşıklık matrisi üzerinde gösterilmiştir.



Şekil 2.26 Karmaşıklık matrisi örneği.

Yanlış Pozitif (FP), modelin gerçekte ilgili sınıfa ait olmayan bir örneği yanlışlıkla o sınıfa aitmiş gibi etiketlemesidir. Yanlış Negatif (FN) ise gerçekte ilgili sınıfa ait olan bir örneğin model tarafından başka bir sınıfa atanması durumunu ifade eder. Buna karşılık köşegen üzerinde yer alan Doğru Pozitif (TP) ve Doğru Negatif (TN) değerleri modelin doğru sınıflandırma başarımını göstermektedir. Bu yapı sayesinde karmaşıklık matrisi yalnızca genel doğruluk oranını yansıtmakla kalmaz, aynı zamanda modelin hangi sınıflarda daha başarılı veya zayıf olduğunun ayrıntılı biçimde analiz edilmesine imkân tanır. Karmaşıklık matrisinde yer alan bu ölçütler doğruluk, kesinlik, duyarlılık ve F1 Skoru gibi başarıım metriklerinin hesaplanmasında temel alınmaktadır. Bu metriklerin matematiksel ifadeleri Eşitlikler D.16 – D.19’da sunulmuştur (Naidu vd. 2023; Yacoubly ve Axman 2020).

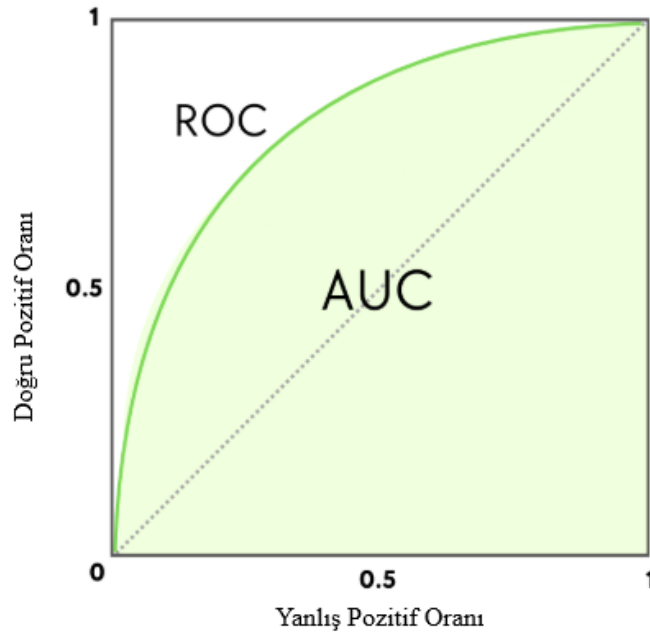
$$\text{Doğruluk (Accuracy)} = \frac{TP + TN}{TP + TN + FP + FN} \quad (\text{D.16})$$

$$\text{Kesinlik (Precision)} = \frac{TP}{TP + FP} \quad (\text{D.17})$$

$$\text{Duyarlılık (Recall)} = \frac{TP}{TP + FN} \quad (\text{D.18})$$

$$\text{F1 Skor (F1Score)} = 2 \times \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (\text{D.19})$$

AUC-ROC eğrisi, sınıflandırma modellerinin ayırt edici gücünü değerlendirmek için yaygın olarak kullanılan bir başarıml metriğidir. ROC eğrisi, farklı eşik değerlerinde modelin doğru pozitif oranı (True Positive Rate - TPR) ile yanlış pozitif oranı (False Positive Rate - FPR) arasındaki ilişkiyi grafiksel olarak göstermektedir (Carrington vd. 2022). Bu eğri altında kalan alan (AUC) değeri, modelin genel sınıflandırma başarısını özetleyen bir ölçüt olup bire yaklaştıkça modelin daha yüksek bir ayırt edici güce sahip olduğunu göstermektedir. Şekil 2.27’de AUC-ROC grafiği örneği verilmiştir.



Şekil 2.27 AUC-ROC grafik örneği.

Genel olarak bu çalışma kapsamında kullanılan tüm başarıml metrikleri, önerilen modellerin sınıflandırma başarısını çok yönlü olarak ortaya koymakta ve modelin yalnızca genel doğruluk oranına değil, aynı zamanda sınıflar arasındaki dengeye, yanlış sınıflandırma eğilimlerine ve ayırt edici gücüne dair derinlemesine bir analiz sunmaktadır. Bu bölümde bahsedilen başarıml değerlendirme metriklerinin birlikte kullanımı, çalışmanın bilimsel geçerliliğini artırarak önerilen modelin güçlü yönlerini ortaya koymaktadır.

3. DENEYSEL SONUÇLAR VE TARTIŞMA

Bu tez çalışmasında klasik CNN modelinin başarımını arttırmak için yeni modeller önerilmiştir. Birinci çalışmada kısıtlı CNN mimarisinde beş farklı model önerilmiştir. Önerilen beş model MNIST, MNIST Fashion ve CIFAR-10 veri kümelerinde tekrarlı bir şekilde uygulanıp sonuçları alınmıştır. Alınan sonuçlar önerilen modellerin hem kendi aralarında hem de klasik CNN modelinin sonuçlarıyla karşılaştırılmıştır. Yapılan değerlendirmeler sonunda literatürdeki diğer çalışmalar ile de karşılaştırılarak güncel çalışmalar ile başarımları değerlendirilmiştir. Bu yapılan incelemeler bölümün devamında detaylı olarak verilmiştir. Çalışmanın devamında ise bu mimari geliştirilerek kısıtlı CNN hesaplamalarında sezgisel optimizasyon algoritmaları kullanılarak OptiConNet ve HyBee-CNN adlarında iki yeni model oluşturulmuştur. OptiConNet modeli MNIST Fashion ve Brain Tumor MRI veri kümelerinde tekrarlı bir şekilde uygulanıp sonuçları alınmıştır. Birinci çalışmada olduğu gibi klasik CNN ile öncelikle modelin doğruluk başarımlarına bakılmıştır. HyBee-CNN modeli de MNIST Fashion veri kümesi üzerinde eğitilip test sonuçları incelenmiştir. Ayrıca modellerin uygulandığı veri kümelerinde yapılan güncel literatür çalışmaları ile karşılaştırmalar yapılarak modelin değerlendirmesi yapılmıştır. Çalışma kapsamında yapılan tüm bu bulgular, ayrı başlıklar altında bu bölüm içerisinde sunulmuştur.

3.1. Kısıtlı CNN Model Sonuçları

Bu çalışmada, C-CNN, S-CNN, D-CNN, M-CNN ve RS-CNN olmak üzere beş farklı Kısıtlı CNN modeli önerilmiştir. Önerilen modellerin başarımlarını değerlendirmesi, yaygın olarak kullanılan MNIST, Fashion-MNIST ve CIFAR-10 veri kümeleri üzerinde gerçekleştirilmiştir. Her bir veri kümesi için önerilen beş model ile eşdeğer mimari ve hiperparametre yapılandırmasına sahip standart bir CNN modeli geliştirilmiş, modeller, doğruluk, F1 Skoru, kesinlik ve duyarlılık metrikleri üzerinden kapsamlı bir şekilde karşılaştırılmıştır. En yüksek genel başarımları gösteren model için ayrıca bir karmaşıklık matrisi hesaplanmış ve sınıflandırma hatalarının dağılımı analiz edilmiştir. Model çıktılarının ayrıntılı yorumlanmasında, sınıf bazlı ayırt edicilik gücünü değerlendirmek amacıyla AUC-ROC eğrileri oluşturulmuş ve sınıf düzeyindeki başarımların farklılıkları incelenmiştir. Son olarak, önerilen modellerin geçerliliğini literatürdeki benzer çalışmalarla kıyaslamak amacıyla, son yıllarda aynı veri kümeleri üzerinde rapor edilen sonuçlarla kapsamlı bir karşılaştırma tablosu hazırlanmış ve elde edilen bulgular tartışılmıştır. Çalışmanın bulguları, her veri kümesine ilişkin olarak ayrı alt başlıklar altında sunulmuştur.

3.1.1. Kısıtlı CNN modelleri MNIST veri kümesi sonuçları

Önerilen Kısıtlı CNN modelleri ilk olarak derin öğrenme uygulamalarının temel veri kümesi olarak görülen MNIST veri kümesine uygulanmıştır. Uygulama yapılırken MNIST veri kümesi için iki tane evrişim katmanı kullanılmıştır. Birinci evrişim katmanında 32 adet 3×3 lük filtre kullanılmıştır ve dolgulama ile adım 1 olarak alınmıştır. Evrişim katmanından sonra maksimum havuzlama yapılmıştır. İkinci evrişim katmanında ise 3×3 'lük 64 filtre kullanılmıştır ve adım-dolgulama 1 olarak belirlenmiştir. Bu yapıda düzleştirme katmanında oluşan vektörler 3136×1 boyutunda oluşacaktır. Tam bağlı katmanda ise üç katman bulunmaktadır. Giriş katmanında 128, gizli katmanda 64 çıkış katmanında ise 10 nöron bulunan her katmanda ReLU aktivasyon fonksiyonu kullanılmıştır. Bu parametreler ile yapılan deneysel uygulamalarda elde edilen sonuçlar, her bir modelin beş kez ayrı ayrı eğitilmesi ve bu eğitimlerden elde edilen test çıktılarının değerlendirilmesiyle oluşturulmuştur. Ayrıca önerilen modeller birbirleri içinde karşılaştırılırken, aynı parametrelere sahip klasik CNN oluşturulup, klasik CNN sonuçları da karşılaştırma tablosuna eklenmiştir. Yapılan beş tekrarlı eğitim ve test süreçleri neticesinde elde edilen karşılaştırma sonuçları Tablo 3.1'de verilmiştir.

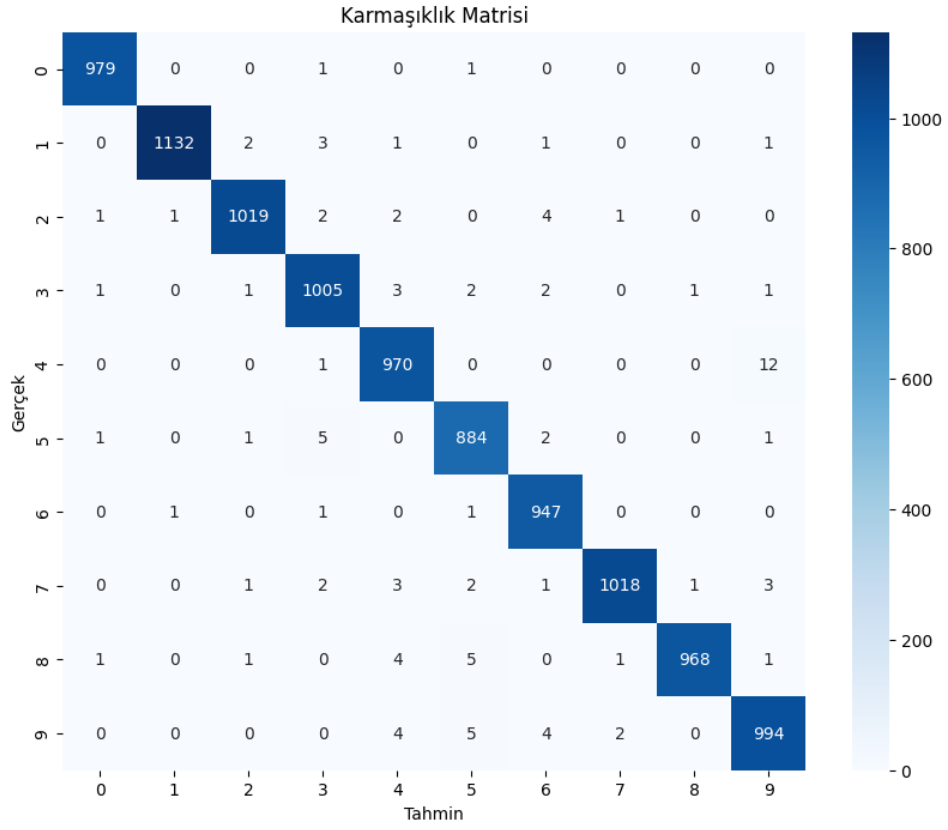
Tablo 3.1 Kısıtlı CNN modellerinin MNIST veri kümesi test sonuçları karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	98,52	0,985	0,9853	0,985
	Ortalama	98,62	0,986	0,9863	0,986
	En İyi	98,92	0,989	0,9892	0,988
C-CNN	En Kötü	98,94	0,989	0,989	0,989
	Ortalama	99,03	0,990	0,990	0,990
	En İyi	99,16	0,991	0,991	0,991
S-CNN	En Kötü	98,33	0,983	0,983	0,983
	Ortalama	98,55	0,985	0,985	0,984
	En İyi	98,73	0,987	0,987	0,987
D-CNN	En Kötü	98,73	0,987	0,987	0,986
	Ortalama	98,92	0,989	0,989	0,989
	En İyi	99,06	0,990	0,990	0,990
M-CNN	En Kötü	98,47	0,984	0,984	0,984
	Ortalama	98,57	0,985	0,985	0,985
	En İyi	98,89	0,988	0,988	0,988
RS-CNN	En Kötü	98,37	0,983	0,983	0,983
	Ortalama	98,43	0,984	0,984	0,984
	En İyi	98,76	0,987	0,987	0,987

Tablo 3.1’de verilen MNIST veri kümesi sonuçları incelendiğinde genel olarak sonuçların oldukça yüksek olduğu görülmektedir. Yapılan tüm testlerde en yüksek doğruluğu %99,16 ile C-CNN elde etmiştir ve D-CNN %99,06 ile ikinci en yüksek doğruluk elde etmiştir. Bu iki model %98,82 doğruluk elde eden Klasik CNN modelinde daha iyi başarımlar göstermiştir. Diğer üç model ise klasik CNN’den daha düşük başarımlar göstermiştir. En iyi

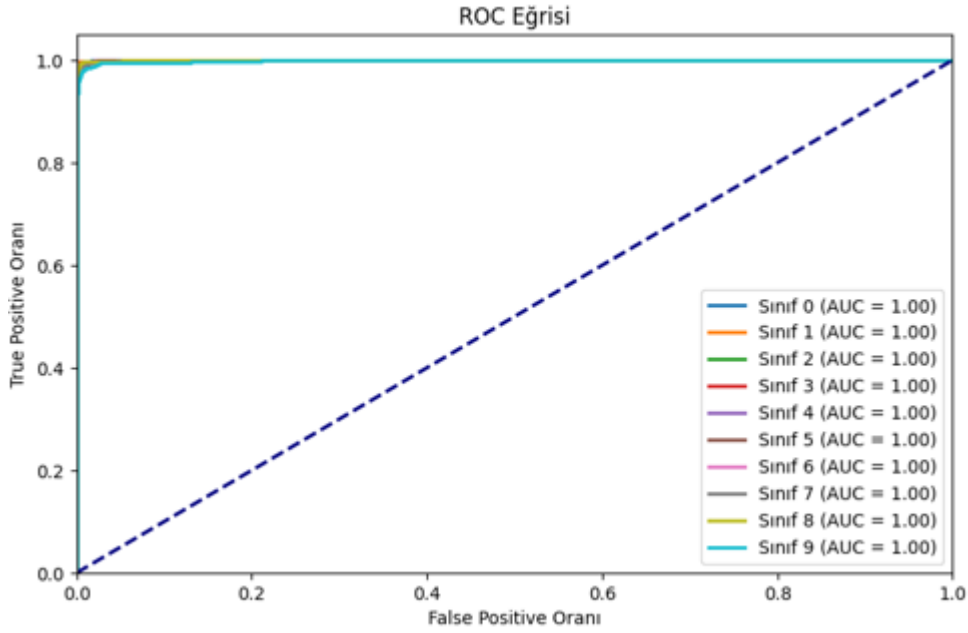
ortalama doğruluğa sahip C-CNN, ortalamada Klasik CNN'den %0,41 en düşük ortalama doğruluk gösteren RS-CNN'den ise %0,60 daha iyi sonuç elde etmiştir. Önerilen beş model ve klasik CNN ile birlikte toplamda altı modelin tekrarlı çalışmalarda hem birbirine çok yakın hem de neredeyse tüm test görsellerine doğru cevap verecek kadar yüksek sonuçlar elde etmesi MNIST veri kümesinin bu değerlendirmeyi yapmak için uygun bir veri kümesi olmadığını göstermektedir. Modellerin birbiri arasında en fazla %0,60'lık bir fark oluşması yeterli bir değerlendirme yapılabilmesinin önüne geçmektedir.

Tablodaki diğer metrikler incelendiğinde, elde edilen F1 Skorları, modellerin hem kesinlik hem de duyarlılık bakımından güçlü ve dengeli bir başarıyı sergilediğini göstermektedir. Özellikle C-CNN ve D-CNN modellerinin sırasıyla 0,9917 ve 0,9907 düzeyinde F1 Skorlarına ulaşması, hata oranının az olduğunu ve genelleme yeteneği kazandığını göstermektedir. Bu modelin aşırı öğrenme olmadan eğitildiğini göstermektedir. Ayrıca C-CNN modelinin kesinlik değerinin 0,9914 olması, yanlış pozitiflerin düşük olduğunu ve modelin pozitif tahminleri yüksek oranda doğru tahmin ettiğini göstermektedir. Genel olarak bu sonuçlar sadece doğruluk metriğinin değil, aynı zamanda sınıflandırma tutarlılığı ve güvenilirliği açısından da dikkate değer bir başarıya sahip olduğunu ortaya koymaktadır. Modelin sınıflandırmasını daha detaylı irdelemek için en yüksek doğruluğa sahip C-CNN modelinin en yüksek başarımadaki sonuçların karmaşıklık matrisi Şekil 3.1'de sunulup devamında elde edilen bulgular verilmiştir.



Şekil 3.1 C-CNN modelinin MNIST veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi

Şekil 3.1’de sunulan karmaşıklık matrisi köşegendeki doğru pozitif değerinin yoğunlukla yüksek olması sınıflandırma başarımını oldukça yüksek ortaya koymaktadır. Önerilen C-CNN modeli özellikle 0, 1, 3 ve 9 numaralı sınıflarda hatasızca yakın tahminler yapmıştır. 4 sınıfını toplam da 12 kez 9 sınıfı olarak tahmin etmiş. Bu hatalı tahmin tablonun en yüksek yanlış negatif değerini oluşturmaktadır ve bunun haricinde 4 sınıfı ile ilgili herhangi bir hatalı tahminde bulunmamıştır. 9 sınıfını tahmin ederken ise 4 kez hatalı bir şekilde 4 sınıfı olarak tahmin etmiştir. Bu durum 4 ile 9 sınıflarını tahmin ederken birbirine benzetip karıştırdığını göstermektedir. Matriste dikkat çeken en yüksek doğru tahmin oranı sıfır sınıfına aittir. Sıfır sınıfına sadece bir kez hatalı tahmin yapmış ve 3 olarak tahmin etmiştir. Genel bir değerlendirme yapıldığında matris doğruluğun yüksek ve yanlış dağılımlarının da çoğunlukla dengeli dağıldığını ve güçlü bir ayırım yapabildiğini gösterirken, toplamdaki yanlış sayısının da istatistiksel olarak oldukça düşük kaldığını göstermektedir. Modelin genel ayırt etme gücünü değerlendirmek amacıyla AUC-ROC eğri grafiği ile Şekil 3.2’de verilmiştir.



Şekil 3.2 C-CNN modeli MNIST veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.2’de verilen grafikte modelin tüm sınıflar için çok yüksek başarımlar göstermesinden kaynaklı çizgiler üst üste binmiş şekilde gösterilmektedir. Bu grafik modelin yüksek duyarlılığa ve yüksek doğruluğa sahip olduğunu göstermektedir.

MNIST veri kümesi derin öğrenme modellerinde genel bir kıyaslama olarak görülse de günümüzde yüksek doğruluk elde eden modeller için birbiri arasında dikkate değer başarımların farklı oluşmamasından kaynaklı farklı veri kümelerinde de test edilme gereksinimini göstermektedir. Bu sebeple çalışmanın devam eden bölümünde önerilen modeller MNIST Fashion ve CIFAR-10 veri kümelerinde de test edilmiştir.

3.1.2. Kısıtlı CNN modelleri MNIST Fashion veri kümesi sonuçları

Önerilen Kısıtlı CNN modelleri MNIST veri kümesine alternatif olarak oluşturulan MNIST Fashion veri kümesinde de eğitilip test edilmiştir. Uygulama yapılırken MNIST veri kümesi için kullanılan filtre sayısı, evrişim katmanı, havuzlama, dolgulama ve adım parametreleri aynı şekilde MNIST Fashion veri kümesi içinde belirlenmiştir. Bu parametreler ile yapılan deneysel uygulamada elde edilen sonuçlar, her bir modelin beş kez ayrı ayrı eğitilmesi ve bu eğitimlerden elde edilen test çıktılarının değerlendirilmesiyle oluşturulmuştur. Ayrıca önerilen modeller birbirleri için de karşılaştırılırken aynı MNIST veri kümesinde yapıldığı gibi, aynı parametrelere sahip klasik CNN oluşturulup, klasik CNN sonuçları da karşılaştırma tablosuna eklenmiştir. Beş tekrar ile yapılan eğitim ve test süreçleri neticesinde

elde edilen karşılaştırma sonuçları Tablo 3.2’de verilmiştir.

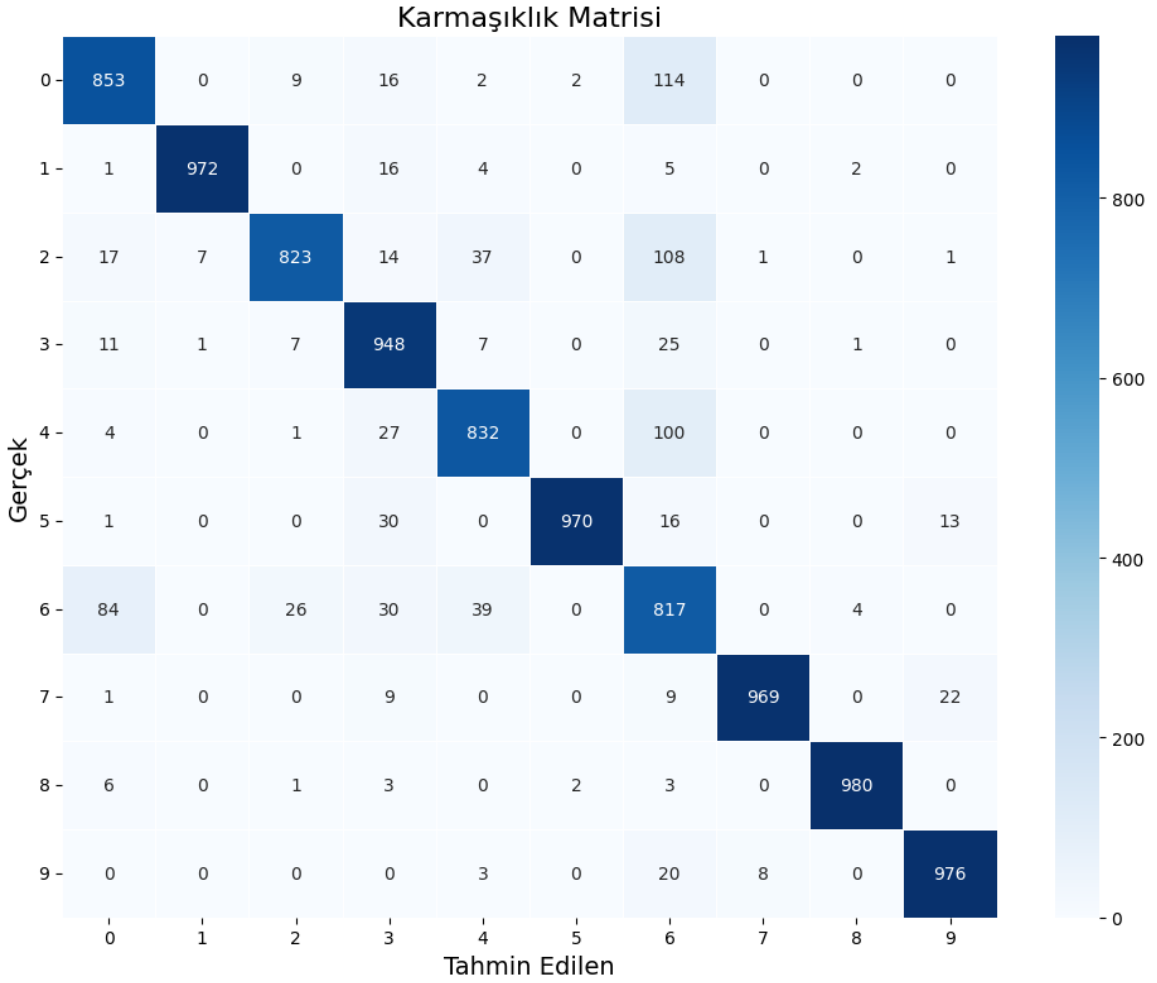
Tablo 3.2 Kısıtlı CNN modellerinin MNIST Fashion veri kümesi test sonuçları karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	89,91	0,898	0,900	0,897
	Ortalama	89,95	0,899	0,900	0,899
	En İyi	89,97	0,900	0,901	0,899
C-CNN	En Kötü	91,39	0,913	0,914	0,914
	Ortalama	91,41	0,914	0,915	0,915
	En İyi	91,43	0,915	0,916	0,915
S-CNN	En Kötü	89,31	0,894	0,895	0,894
	Ortalama	89,69	0,899	0,900	0,897
	En İyi	90,03	0,900	0,903	0,900
D-CNN	En Kötü	91,51	0,916	0,916	0,915
	Ortalama	91,80	0,918	0,917	0,917
	En İyi	92,11	0,921	0,922	0,921
M-CNN	En Kötü	90,87	0,908	0,909	0,908
	Ortalama	90,92	0,909	0,910	0,909
	En İyi	90,94	0,910	0,911	0,910
RS-CNN	En Kötü	88,31	0,885	0,887	0,884
	Ortalama	88,39	0,887	0,889	0,885
	En İyi	88,44	0,908	0,909	0,907

Tablo 3.2’de verilen MNIST Fashion veri kümesi sonuçları incelendiğinde genel olarak doğruluk sonuçlarının %88 ila %92 aralığında olduğu görülmektedir. Yapılan tüm testlerde en yüksek doğruluğu %92,11 ile D-CNN elde etmiştir. D-CNN’den sonra en yüksek doğruluğu

%91,43 ile C-CNN modeli elde etmiştir. S-CNN ve M-CNN modelleri ise sırasıyla %91,43 ve %90,94'lük doğruluk elde ederek, %89,97 doğruluk elde eden Klasik CNN'den daha yüksek başarımlar göstermiştir. Bu sonuçlara göre klasik CNN sadece RS-CNN modelinden %1,56'lık bir fark ile daha yüksek doğruluk elde etmiştir. Diğer önerilen dört model CNN'den daha iyi sonuç elde etmiştir. En yüksek doğruluk elde eden D-CNN ile en düşük doğruluk elde eden RS-CNN arasında %3,39'luk bir fark oluşmuştur. Modellerin başarımlar dağılımı, MNIST'e kıyasla MNIST Fashion veri kümesinde daha yüksek değişim göstermiştir. Bu nedenle modellerin ayırt edici yeteneklerinin değerlendirilmesinde MNIST Fashion sonuçlarının daha güvenilir bir kriter oluşturduğu değerlendirilmektedir.

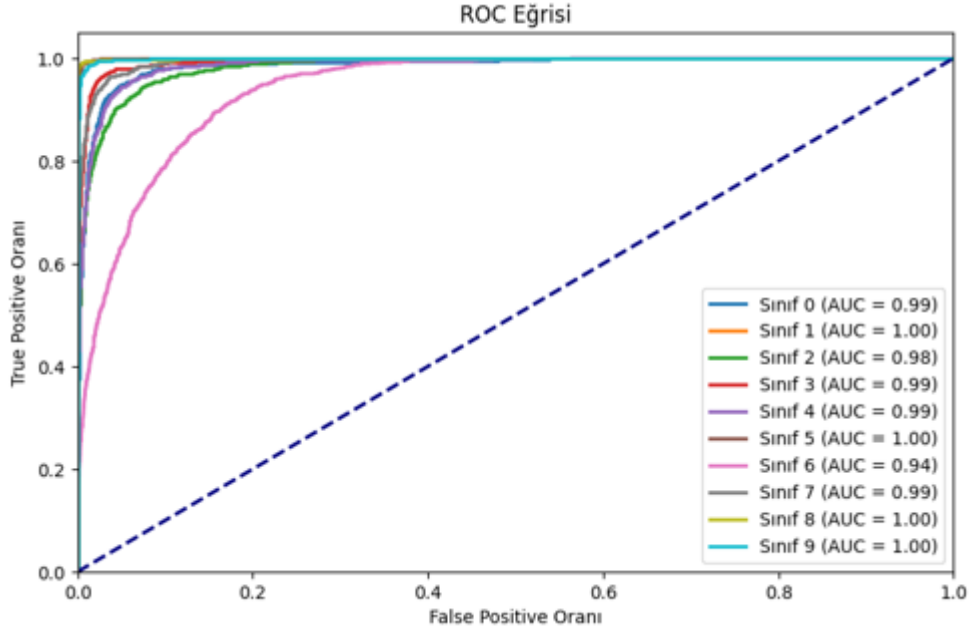
Doğruluk metriği dışında kalan F1 Skoru, kesinlik ve duyarlılık değerleri incelendiğinde, modellerin genel olarak birbirine yakın fakat anlamlı farklılıklar sergiledikleri görülmektedir. Elde edilen sonuçlara göre. D-CNN modeli bu üç metrikte de en yüksek değerlere ulaşmıştır. Ortalama F1 Skoru 0,918, kesinlik 0,917 ve duyarlılık 0,917 olan D-CNN hem kesinlik hem de duyarlılık açısından dengeli bir yapı göstermektedir. Bu durum modelin doğru sınıflandırma yaparken yanlış pozitif ve yanlış negatif oranlarını düşük seviyede tutabildiğini göstermektedir. RS-CNN modeli bu metrikler bakımından en düşük değerlere ulaşmıştır. Ortalama F1 Skoru 0,887, kesinlik 0,889 ve duyarlılık 0,885 olan model özellikle duyarlılık açısından belirgin bir düşüş göstermektedir. Genel olarak değerlendirildiğinde. F1 Skor, kesinlik ve duyarlılık açısından D-CNN ve C-CNN modellerinin açık bir üstünlüğü söz konusudur. Bu modeller yalnızca doğru sınıflandırma oranlarını artırmakla kalmamış, aynı zamanda yanlış sınıflandırma oranlarını da minimize ederek veri kümesindeki örüntüleri daha etkin bir biçimde yakalamışlardır. Modelin doğru ve yanlış sınıflandırmalarını daha detaylı incelemek için en yüksek doğruluk elde eden D-CNN modelinin elde ettiği en yüksek doğruluğa ait test sonuçları ile karmaşıklık matrisi oluşturulup Şekil 3.3'te sunulmuştur.



Şekil 3.3 D-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi

Şekil 3.3'te sunulan karmaşıklık matrisi köşegendeki doğru pozitif değeri 1 (pantolon), 3 (elbise), 5 (sandalet), 7 (spor ayakkabı), 8 (çanta), 9 (bilek boy bot) sınıflarında, diğer sınıflara kıyasla daha yüksek yoğunluğa sahip olduğu görülmektedir. Bu sınıflarda model daha yüksek doğruluk değerine ulaşmıştır. En yüksek doğruluk elde ettiği 8 (Çanta) sınıfını tahmin ederken 1000 veriden sadece 20 tanesine hatalı cevap vermiştir. Hatalı cevap verdiği 20 cevaptan en çok 0 (t-shirt) sınıfı ile karıştırıp 6 kez hatalı cevap vermiştir. En düşük doğruluk elde ettiği 6 (gömlek) sınıfında verilerin %81,7'sine doğru tahminde bulunmuştur. Bu sınıfa yaptığı tahminlerin hatalı cevapları tek bir sınıfa kümelenmemiştir. Gömlek en çok 0 (t-shirt) sınıfında 84 hatalı tahminde bulunurken, ikinci olarak ise 4 (ceket) sınıfını 39 kez tahmin etmiştir. Karmaşıklık matrisinde dikkat çeken başka bir durum ise 6 (gömlek) sınıfı, diğer sınıflar için en çok hatalı tahmin yapılan sınıf olmuştur. Toplamda 359 kez başka sınıflar 6 (gömlek) sınıfı olarak tahmin edilmiştir. Bu yanlış tahminlerin dağılımı 0 (t-shirt), 2 (kazak) ve 4 (ceket)

sınıflarında sırasıyla 114, 108 ve 100 olarak matrisin geneldeki hatalı tahmin dağılımının çok üzerinde çıkmıştır. Bu durumun tam tersi ise 1 (pantolon) sınıfında olmuştur. Diğer sınıflara sadece bir kez hatalı olarak 1 (pantolon) tahmininde bulunmuştur. Ortaya çıkan bu durum sadece doğruluk üzerinden değil, modelin başarımlarını anlamak için karmaşıklık matrisinin kullanılması gerekliliğini ortaya çıkarmıştır. Şöyle ki, 8 (çanta) sınıfı %98’lik bir doğruluk elde etmiştir ancak 12 kez farklı sınıflara 0 (çanta) tahmininde bulunulmuştur diğer taraftan %97,2’lik bir doğruluğa sahip olan 1 (Pantolon) sınıfı sadece 1 kez farklı sınıflara hatalı cevap olarak verilmiştir. Elde edilen bu sonuçlar, modelin 8 sınıfı (çanta) için yüksek duyarlılık, ancak görece daha düşük kesinlik değerine sahip olduğunu göstermektedir. Bu, çantaların çoğunun başarıyla sınıflandırıldığını, ancak modelin bazen diğer sınıflara ait örnekleri de “çanta” olarak yanlış etiketleyebildiğini (düşük kesinlik) ifade etmektedir. Buna karşın, 1 (pantolon) sınıfı için kesinlik değeri oldukça yüksek olup, modelin bu sınıfı tahmin etmesi durumunda neredeyse her zaman doğru sonuç verdiği gözlenmiştir. Bununla birlikte, bu sınıfın duyarlılık değeri çantadan daha düşüktür; yani gerçek pantolon örneklerinin bir kısmı kaçırılmakta olup, model bu sınıfa karşı daha az duyarlı davranmaktadır. Dolayısıyla, sınıf 1’in (pantolon) sınıf 8’e (çanta) kıyasla daha düşük doğruluk değil, daha düşük duyarlılık, ancak daha yüksek kesinlik sergilediği sonucuna varılabilir. Bu durum, değerlendirme sürecinde genel doğruluk yerine sınıf bazlı kesinlik duyarlılık gibi metriklerin kullanılmasının kritik önemini vurgulamaktadır. Ayrıca sınıfların ayırt edicilik başarımlarını farklı bir grafik ile görselleştirebilmek için Şekil 3.4’te modelin AUC-ROC grafiği sunulmuştur.



Şekil 3.4 D-CNN modeli MNIST Fashion veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.4'te sunulan grafikte en dikkat çekici durum pembe renk ile gösterilen 6(gömlek) sınıfının en düşük eğri olmasıdır. Bu durum karmaşıklık matrisi incelemesinde de vurgulanan 6(gömlek) sınıfının ayırt ediciliğinin düşük olmasından kaynaklanmaktadır. Ayırt ediciliği düşük ikinci sınıf ise 2 (kazak) sınıfı olmuştur. Yüksek ayırt edicilik yeteneğine sahip sınıfların ROC eğrileri, TPR'nin ve FPR'nin çok düşük değerlerinde hızla 1'e ulaşması nedeniyle birbiriyle üst üste binmekte ve görsel olarak ayrımları net şekilde ortaya çıkmamaktadır.

MNIST Fashion veri kümesinde modeller önce kendi içerisine kıyaslanıp sonra başarılı başarımlarını sınıf bazında incelendikten sonra son olarak literatürde yapılan çalışmalar ile karşılaştırılıp Tablo 3.3'de sunulmuştur.

Tablo 3.3 D-CNN modelinin MNIST Fashion veri kümesi için literatürdeki çalışmaları ile karşılaştırılması.

Yazar / Yıl	Model	Doğruluk
(Greeshma ve Sreekumar 2019)	HOG + SVM	%85,50
(Bhatnagar vd. 2017)	CNN2	%91,17
(Venkataravanappa vd. 2024a)	Resnet50	%90,25
(Sun vd. 2023)	Tiny VGG (Max pooling)	%88,21
	Min Pooling	%87,78
(López-Ortiz vd. 2024)	ESN	%88,10
	MR ESN	%89,12
(Huang vd. 2021)	Laser based RC	%85,46
(Haider vd. 2023)	COVNet	%88,40
(Sun vd. 2023)	MADPL-net	%90,11
(Erkoç ve Eskil 2023)	Unsupervised Filter Learning Method	%91,24
Önerilen Model	D-CNN	%91,80

Önerilen modellerden en yüksek doğruluk elde eden D-CNN modeli literatürde özellikle son yıllarda yapılan çalışmalar ile doğruluk değerleri karşılaştırılmıştır. Yapılan karşılaştırma sonucunda D-CNN modeli literatürden alınan örnekleme göre iyi bir başarıyı göstermiştir. Bu karşılaştırmaya dayanarak önerilen model literatürde kullanılabilir bir alternatif oluşturmaktadır.

3.1.3. Kısıtlı CNN modelleri CIFAR-10 veri kümesi sonuçları

Önerilen Kısıtlı CNN modelleri MNIST ve MNIST Fashion veri kümelerinden sonra renkli görseller içeren ve diğer iki veri kümesine göre daha zor bir sınıflandırma problemi olan CIFAR-10 veri kümesine uygulanmıştır. Uygulama yapılırken CIFAR-10 veri kümesi için iki tane evrişim katmanı kullanılmıştır. Birinci evrişim katmanında 32 adet 3×3 lük filtre

kullanılmıştır ve dolgulama ve adım 1 olarak alınmıştır ve devamında maksimum havuzlama yapılmıştır. İkinci evrişim katmanında ise 3×3 'lük 64 filtre kullanılmıştır, adım kaydırma ve dolgulama 1 olarak belirlenmiştir. Bu yapıda düzleştirme katmanında oluşan vektörler üç kanallı RGB görüntü olduğundan 4096×3 boyutlu oluşacaktır. Tam bağlı katmanda ise üç katman bulunmaktadır. Giriş katmanında 128, gizli katmanda 64 çıkış katmanında ise 10 nöron ve her katmanda ReLU aktivasyon fonksiyonu kullanılmıştır. Bu parametreler ile yapılan deneysel uygulamada elde edilen sonuçlar, her bir modelin beş kez ayrı ayrı eğitilmesi ve bu eğitimlerden elde edilen test çıktılarının değerlendirilmesiyle oluşturulmuştur. Ayrıca, önerilen modellerin kendi aralarındaki başarımlarını ortaya koymak amacıyla başarımları doğru bir şekilde değerlendirebilmek için aynı parametre değerlerine sahip klasik bir CNN modeli oluşturulup önerilen modellerle karşılaştırılmıştır. Uygulamada beş kez eğitilen modellerin test sonuçları en iyi, ortalama ve en kötü başarımları Tablo 3.4'te sunulmuştur.

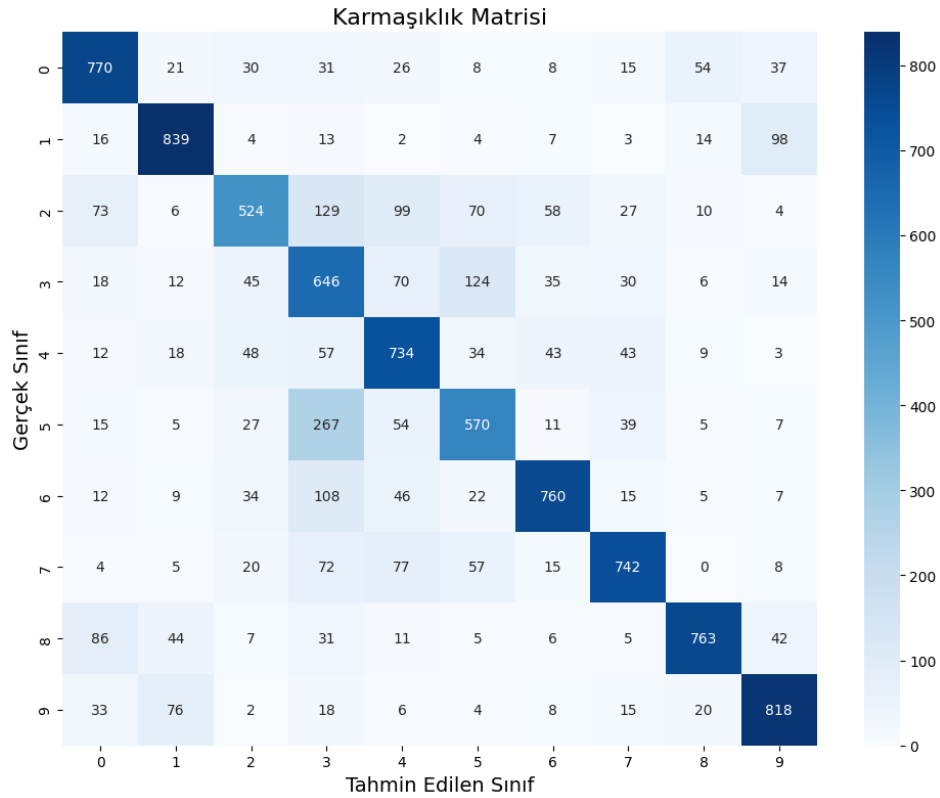
Tablo 3.4 Kısıtlı CNN modellerinin CIFAR-10 veri kümesi test sonuçlarının karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	65,14	0,653	0,658	0,652
	Ortalama	65,59	0,667	0,660	0,658
	En İyi	66,02	0,669	0,662	0,660
C-CNN	En Kötü	71,01	0,711	0,718	0,710
	Ortalama	71,22	0,713	0,719	0,712
	En İyi	71,43	0,715	0,720	0,714
S-CNN	En Kötü	67,21	0,672	0,680	0,672
	Ortalama	67,71	0,677	0,686	0,676
	En İyi	67,98	0,681	0,692	0,680
D-CNN	En Kötü	71,12	0,708	0,712	0,712
	Ortalama	71,44	0,713	0,720	0,714
	En İyi	71,69	0,719	0,728	0,717
M-CNN	En Kötü	68,69	0,688	0,697	0,687
	Ortalama	68,94	0,690	0,703	0,689
	En İyi	69,19	0,695	0,709	0,692
RS-CNN	En Kötü	69,18	0,691	0,699	0,692
	Ortalama	69,97	0,701	0,712	0,709
	En İyi	70,55	0,711	0,717	0,712

Tablo 3.4'te sunulan verilere göre modellerin doğruluk değerleri karşılaştırıldığında, klasik CNN modelinin diğer modellere kıyasla daha düşük bir başarımla sergilediği görülmektedir. CNN modelinin ortalama doğruluğu %65,59 iken, bu oran diğer önerilen modellerde belirgin bir fark ile daha yüksek olduğu görülmektedir. Özellikle C-CNN (%71,22) ve D-CNN (%71,44) modelleri, en yüksek ortalama doğruluk oranlarına ulaşmışlardır. Bu iki

modelden sonra RS-CNN modeli de %69,97'lik ortalama doğruluk başarımı göstermiştir. M-CNN (%68,94) ve S-CNN (%67,71) modelleri ise bu iki modelin ardından gelmiştir. En iyi başarımlara bakıldığında da sıralama aynı olmuştur. En başarılı başarımları gösteren D-CNN modeli, klasik CNN'den %5,67'lik bir fark ile daha iyi başarımları elde etmiştir. Bu oluşan fark MNIST veri kümesinde %0,44, MNIST Fashion veri kümesinde %2,14 olmuştur. Ayrıca diğer veri kümelerinde bazı modeller klasik CNN'in gerisinde kalırken CIFAR-10 veri kümesinde önerilen beş kısıtlı CNN modeli klasik CNN'den daha yüksek doğruluk değeri elde etmiştir.

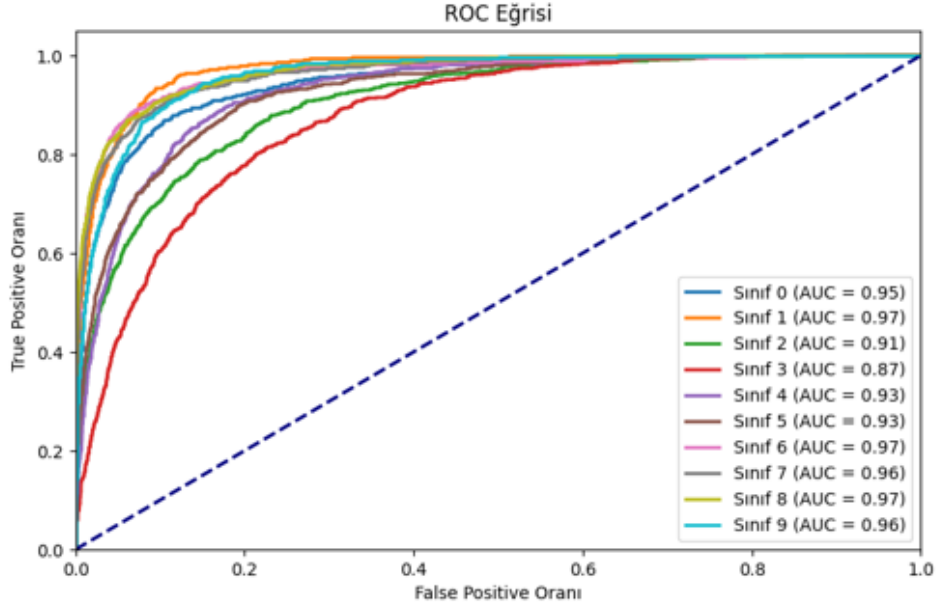
F1 Skoru, duyarlılık ve kesinlik değerleri incelendiğinde, modeller arasındaki farkların doğruluk metriğiyle paralel bir eğilim sergilediği görülmektedir. C-CNN ve D-CNN modelleri hem ortalama F1 Skoru hem de duyarlılık ve kesinlik değerleri bakımından diğer modellere göre daha yüksek sonuçlar elde etmiştir. Doğruluk metriğinde olduğu gibi önerilen modeller F1 Skoru, duyarlılık ve kesinlik metriklerinde klasik CNN'den daha yüksek başarımları göstermişlerdir. Bu sonuçlar söz konusu modellerin yalnızca genel doğruluk açısından değil, aynı zamanda pozitif örnekleri doğru şekilde tanımlama ve yanlış pozitifleri minimize etme açısından da daha dengeli ve güvenilir sonuçlar elde ettiğini göstermektedir. Tablo 7'de verilen bu bilgilerden en yüksek başarımları sahip D-CNN modelinin 5 tekrarından en iyi doğruluk veren sonucundan karmaşıklık matrisi oluşturulup, Şekil 3.5'de sunulmuştur.



Şekil 3.5 D-CNN modelinin CIFAR-10 veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.

Şekil 3.5’de sunulan karmaşıklık matrisi köşegendeki doğru tahmin değeri 1 (otomobil), 0 (uçak), 9 (kamyon), 6 (kurbağa), 8 (gemi), 9 (at) sınıflarında modelin ortalama doğruluğundan daha yüksek değer elde ettiği görülmektedir. En yüksek doğruluğa sahip 1 (otomobil) sınıfında görsellerin %83,9’u doğru tahmin edilmiştir. Bu sınıf en çok 9 (kamyon) sınıfı ile karıştırılmıştır. Otomobil sınıfına yapılan 161 hatalı tahminin 98 tanesi 9 (kamyon) sınıfı olarak tahmin edilmiştir. En düşük doğruluğa sahip 2 (Kuş) sınıfı için yapılan tahminlerin %52,4’ü doğru tahmin edilmiştir. Kuş sınıfına yapılan yanlış tahminler incelendiğinde 1(otomobil) ve 9 (kamyon) tahminleri çok az yapılırken, 2 (kuş) sınıfına yapılan 576 hatalı tahminin 227’si 3 (kedi) ve 4 (geyik) olarak tahmin edilmiştir. Karmaşıklık matrisinde dikkat çeken en fazla yapılan hatalı tahmin 3 (kedi) ve 5 (köpek) sınıfları arasında olmuştur. Köpek sınıfına hatalı olarak 267 kez kedi tahmini yapılırken, kedi sınıfına ise 124 kez köpek tahmini yapılmıştır. Bu sonuçlar modelde birbirine en çok karıştırılan sınıflar olduğunu göstermektedir. Birbirine en az karıştırılan sınıflar ise 7 (at) ile 8 (gemi) sınıfları olmuştur. Model test işleminde hiçbir at görseline gemi demezken, gemi görsellerinin de sadece 5 tanesine at tahmininde bulunmuştur. Buradaki dikkat çekici durum ise karmaşıklık matrisinde 0 değerine sahip tek hücre 7 (at) sınıfına yapılan 8 (gemi) tahmini sayısını içeren hücre olmuştur. Bütün sınıflar içerisinde başka sınıflara en çok hatalı olarak tahmin yapılan 3 (kedi) sınıfı olmuştur. Kedi sınıfı toplamda 731

kez farklı sınıflara tahmin yapılarak ayırt ediciliği en düşük sınıf olmuştur. Sınıfların bu ayırt edicilik başarımlarını farklı bir grafik ile görselleştirebilmek için Şekil 3.6’da modelin AUC-ROC grafiği sunulmuştur.



Şekil 3.6 D-CNN modeli CIFAR-10 veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.6’da verilen grafikte önceki yapılan MNIST ve MNIST Fashion veri kümelerine göre daha belirgin birbirinden ayrılan eğriler olduğu görülmektedir. Kırmızı ile gösterilen 3 (kedi) sınıfı modelin ayırt etmekte en çok güçlük çektiği sınıf olduğu görülmektedir. 1 (otomobil), 6 (kurbağa) ve 9 (gemi) sınıfları ise ayırt edilmesi yüksek olan sınıflar olarak görülmektedir.

CIFAR-10 veri kümesinde modeller önce kendi içerisine karşılaştırılıp sonra başarılı başarımlarını sınıf bazında incelendikten sonra son olarak literatürde yapılan çalışmalar ile karşılaştırılıp, Tablo 3.5’te verilmiştir.

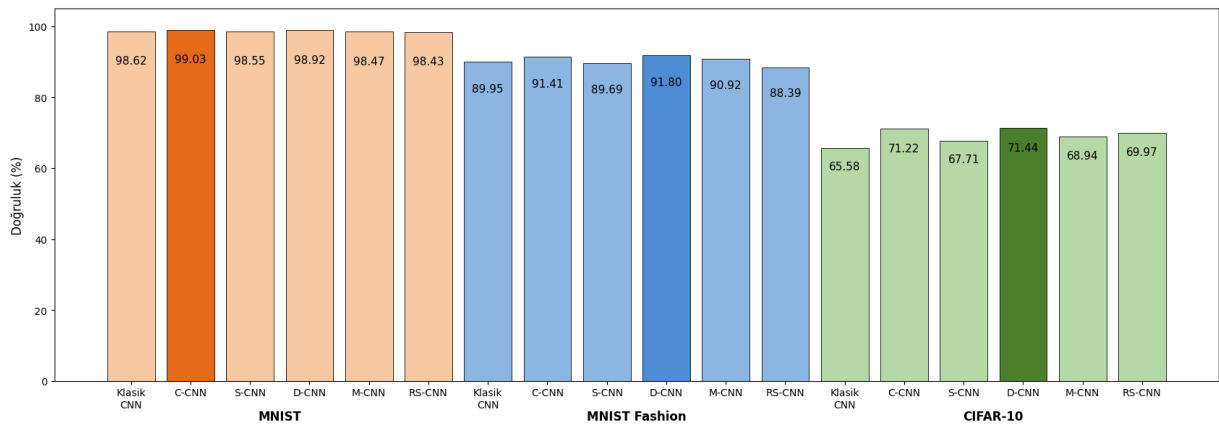
Tablo 3.5 D-CNN modelinin CIFAR-10 veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.

Çalışma	Model	Doğruluk
(Grønningsæter vd. 2024)	CTM	%60,70
(Azam ve Akhtar 2024)	Simple MLP	%39,50
	ConNet (Large)	%71,00
	ConvKANLinear	%61,60
	KConvKAN	%62,60
(Li vd. 2024)	LENET-5	%55,16
	ALEXNET	%69,75
(Doğan 2023)	Lenet-5 2*2 AVG pooling	%66,95
(Aslam ve Nassif 2023)	VGG-16	%61,00
	VGG-19	%60,00
(Cui vd. 2023)	Tesla (10 Epok)	%66,40
	FrePro(10 Epok)	%65,50
	MTT (10 Epok)	%65,30
Önerilen Model	D-CNN	%71,44

Önerilen modellerden en yüksek doğruluk elde eden D-CNN modeli literatürde özellikle son yıllarda yapılan çalışmalardaki ile doğruluk değerleri ile karşılaştırılmıştır. Yapılan karşılaştırma incelendiğinde özellikle 2024 yılında Justin Cui ve arkadaşlarının yaptığı çalışma uygulanan tur (epok) sayısı, bu çalışmada tercih edilen tur (epok) sayısına eşit bulunmaktadır bu sebeple iki çalışmanın karşılaştırmasını dikkat çekicidir. Justin Cui ve arkadaşlarının yaptığı çalışmada elde ettikleri en yüksek doğruluk değeri %66,4 olurken, önerilen D-CNN modeli %71,44 doğruluk elde ederek iki çalışma arasında %5'lik bir fark oluşmuştur. Yapılan karşılaştırma sonucunda D-CNN modeli literatürden alınan örnekleme göre %71,00 ile en yüksek doğruluğa sahip çalışmadan %0,44 daha iyi bir doğruluk göstermiştir.

3.1.4. Kısıtlı CNN modelleri genel değerlendirme

Kısıtlı CNN çalışması kapsamında oluşturulan modeller üç farklı veri kümesinde tekrarlı bir şekilde eğitilip sonuçları detaylı bir şekilde incelenmiştir. Genel olarak sonuçlara bakıldığında MNIST veri kümesinde D-CNN ve C-CNN modeli klasik CNN'den daha yüksek başarımlar göstermiştir. MNIST Fashion veri kümesinde D-CNN, C-CNN, M-CNN ve S-CNN modelleri klasik CNN'den daha yüksek başarımlar göstermiştir. CIFAR-10 veri kümesinde ise önerilen modellerin tamamı klasik CNN'den daha yüksek doğruluk elde etmiştir. Uygulanan bu veri kümelerinde doğruluk metriği üzerinden tek bir grafik üzerinde Şekil 3.7'de özetlenerek sunulmuştur.



Şekil 3.7 Kısıtlı CNN modellerinin farklı veri kümelerinde doğruluk metriği için çubuk grafiği

Şekil 3.7'de verilen grafikte her bir veri kümesi ayrı renklerde gösterilmiştir. MNIST turuncu, MNIST Fashion mavi, CIFAR-10 ise yeşil renkte çizilerek veri kümeleri ayrımları daha belirgin hale getirilmiştir. Ayrıca her bir veri kümesinde en başarılı başarımları gösteren model koyu renkte belirtilerek başarılı modeller vurgulanmıştır. Önerilen kısıtlı CNN modelleri için yapılan tüm bu uygulamalar neticesinde dikkate değer iki farklı sonuç elde edilmiştir. Bunlar:

- Problem zorlaştıkça kısıtlı CNN modellerinin doğruluğu daha çok yükselmektedir. MNIST veri kümesinde C-CNN modeli klasik CNN'den %0,6'lık daha yüksek bir doğruluk elde etmiştir. MNIST Fashion veri kümesinde D-CNN modeli klasik CNN'den %1,56'lık daha yüksek bir doğruluk elde etmiştir. CIFAR-10 veri kümesinde ise D-CNN modeli klasik CNN'den %5,67'lik daha yüksek bir doğruluk elde etmiştir. Bu sıralamaya dikkat edildiğinde model gittikçe zorlaşırken klasik CNN ile arasındaki fark yükselmiştir. Bu durum önerilen modellerin zor

problemlerde daha yüksek verimli sonuçlar elde ettiğini göstermektedir.

- Her bir veri kümesindeki en başarılı sınıflara dikkat edildiğinde, önerilen modellerin öne çıktığı görülmektedir. MNIST veri kümesinde en başarılı iki model sırasıyla C-CNN ve D-CNN modelleri olmuştur. MNIST Fashion veri kümesinde en başarılı iki model sırasıyla D-CNN ve C-CNN modelleri olmuştur. CIFAR-10 veri kümesinde ise en başarılı iki model sırasıyla yine D-CNN ve C-CNN modelleri olmuştur. Bu durumda D-CNN ve C-CNN modellerinin diğerlerinden farkı incelenmiştir. İkinci bölümde açıklanan kısıtlı atama yapıları incelendiğinde, bu iki modelin diğer modellerden farklı olarak seçilen sınıflar üzerinde yalnızca çıkarma işlemi gerçekleştirdiği, buna karşılık diğer modellerin ise toplama ya da toplama/çıkarma şeklinde karma işlemler yürüttüğü görülmektedir. Bu durumda kısıtlı CNN modellerinin fark işlemlerinde daha etkili olduğu görülmektedir. Bu sebeple çalışmanın devamında oluşturulacak modeller sadece C-CNN ve D-CNN modellerinin alt yapısıyla üretilen modeller olmuştur.

Çalışma kapsamında önerilen kısıtlı modellerin başarımlarını değerlendirme, farklı veri kümeleri üzerinde çok sayıda metrik kullanılarak kapsamlı bir şekilde gerçekleştirilmiştir. Bunların yanı sıra, model kısıtlamalarının eğitim süresi üzerindeki etkisini nicel olarak değerlendirmek amacıyla her bir modelin eğitim süresi ayrı ayrı kaydedilmiş ve sonuçlar Tablo 3.6’da saniye cinsinden sunulmuştur.

Tablo 3.6 Kısıtlı CNN çalışma sürelerinin saniye cinsinden karşılaştırılması.

Veri Kümesi	D-CNN	C-CNN	M-CNN	S-CNN	RS-CNN	CNN
MNIST	853	858	904	863	897	728
MNIST Fashion	973	945	979	942	953	834
CIFAR-10	1123	1111	1162	1106	1125	943

Tablo 3.6’deki süreler incelendiğinde önerilen modeller hep birbirlerine yakın sürede eğitimleri tamamlamışlardır. Önerilen modeller içerisinde M-CNN modeli her zaman en yavaş çalışan model olmuştur. Klasik CNN ise %10 ile %15 aralığında önerilen modellerden zaman tasarrufu sağlamaktadır. Önerilen modeller zaman açısından bir kayıp oluştursa da tur (epok)

sayısı arttığında bu aradaki farkın azalacağı öngörülmektedir. Kısıtlı CNN modeli düşünüldüğünde başlangıçta bir ek hesaplama yükü oluşsa da sonrasında klasik CNN gibi çalıştığı için devamında ek bir yavaşlama olmayacaktır ve klasik CNN'e kıyasla daha yüksek doğruluk elde edebildiği için, modelin eğitim sürecinde hedeflenen eşiğe daha erken ulaşacağı ve dolayısıyla zaman farkının azalacağı öngörülmektedir.

3.2. OptiConNet Model Sonuçları

OptiConNet, çalışmada önerilen kısıtlı CNN modellerinden D-CNN ve C-CNN'in temel çalışma algoritmasının. SCO sezgisel optimizasyon algoritmasıyla karma bir yapı oluşturularak geliştirilmiş yeni bir modelidir. Modelin ilk olarak literatürdeki geçerliliğini değerlendirmek için model MNIST Fashion veri kümesinde eğitilip test edilmiştir. MNIST Fashion için beş kez tekrarlı eğitim ve test yapıp en iyi, ortalama ve en yüksek başarımları kayıt altına alınmıştır. Bu sonuçlar aynı parametreler ile çalışan klasik CNN ile karşılaştırılmıştır. Karşılaştırma sonrasında modelin güçlü ve zayıf yönleri karmaşıklık matrisi ve AUC-ROC grafiği kullanılarak incelenmiştir. Model tüm yönlerinden incelendikten sonra literatürdeki diğer çalışmalar ile karşılaştırma yapılarak modelin değerlendirmesi yapılmıştır. Modelin ayrıca gerçek hayat problemlerindeki başarımını incelemek için Brain Tumor MRI veri kümesi üzerinde eğitilip test edilmiştir. Test sonuçları MNIST Fashion veri kümesinde olduğu gibi klasik CNN ile karşılaştırılıp, karmaşıklık matrisi ve AUC-ROC grafiği çıkartılmıştır. Son olarak elde edilen sonuçlar Brain Tumor MRI veri kümesinde daha önceden yapılmış literatürdeki çalışmalar ile karşılaştırılmıştır. Bu bölümün devamında uygulama ve sonuçlar ile ilgili detaylar verilmiştir.

3.2.1. OptiConNet modeli MNIST Fashion veri kümesi sonuçları

Önerilen OptiConNet modeli ilk olarak MNIST Fashion veri kümesinde eğitilip test edilmiştir. Uygulama yapılırken MNIST Fashion veri kümesi için iki tane evrişim katmanı kullanılmıştır. Birinci evrişim katmanında 32 adet 3×3 lük filtre kullanılmıştır. Dolgulama ve adım 1 olarak alınıp maksimum havuzlama yapılmıştır. İkinci evrişim katmanında ise 3×3 'lük 64 filtre, adım ve dolgulama 1 olarak belirlenmiştir. Bu katmanın devamında maksimum havuzlama kullanılmıştır. Bu yapıda düzleştirme katmanı sonrasında elde edilen vektörler 3136×1 boyutunda oluşacaktır. Tam bağlı katmanda ise üç katman bulunmaktadır. Giriş katmanında 128, gizli katmanda 64, çıkış katmanında ise 10 nöron ve her katmanda ReLU aktivasyon fonksiyonu kullanılmıştır. Ayrıca modelin içerisinde yer alan SCO algoritması için maksimum iterasyon 100, maksimum başarısız güncelleme eşiği 5, b parametresi 2,4, arama

uzayının minimum ve maksimum sınırları 0-59999 olarak belirlenmiştir. Bu parametreler ile yapılan deneysel uygulamada elde edilen sonuçlar, her bir modelin beş kez ayrı ayrı eğitilmesi ve bu eğitimlerden elde edilen test çıktılarının değerlendirilmesiyle oluşturulmuştur. Ayrıca önerilen OptiConNet modeli ile aynı parametrelere sahip klasik CNN oluşturulup, klasik CNN ve bir önceki çalışmadaki D-CNN sonuçları ile karşılaştırılmıştır. Tekrarlı eğitim ve test süreçleri neticesinde elde edilen metrikler Tablo 3.7’de özet olarak verilmektedir.

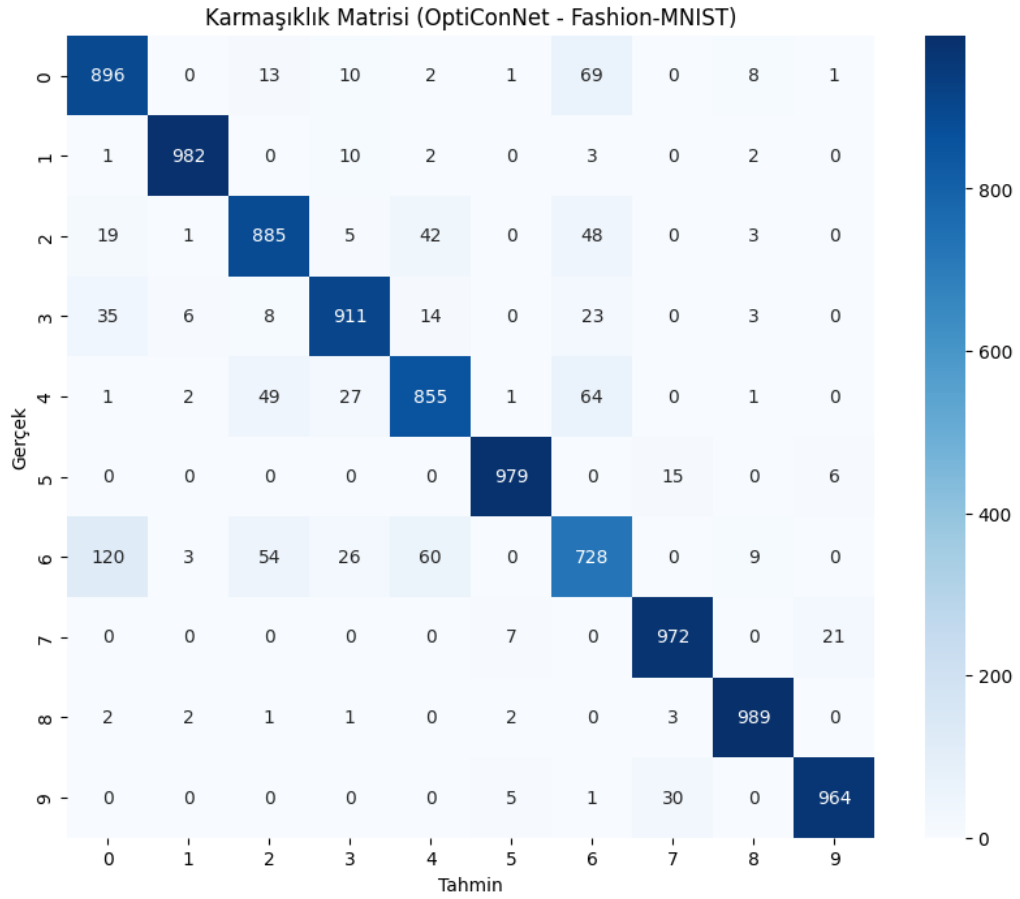
Tablo 3.7 OptiConNet modeli MNIST Fashion veri kümesi test sonuçlarının karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	88,91	0,881	0,890	0,867
	Ortalama	89,65	0,899	0,875	0,899
	En İyi	89,97	0,900	0,901	0,899
D-CNN	En Kötü	91,51	0,916	0,916	0,915
	Ortalama	91,80	0,918	0,917	0,917
	En İyi	92,11	0,921	0,922	0,921
OptiConNet	En Kötü	91,85	0,918	0,919	0,915
	Ortalama	92,25	0,928	0,922	0,920
	En İyi	92,55	0,929	0,926	0,922

Tablo 3.7’de verilen veriler incelendiğinde, klasik CNN modeli ve D-CNN ile önerilen OptiConNet modelinin doğruluk değerleri karşılaştırıldığında, OptiConNet’in en kötü, ortalama ve en iyi olmak üzere her üç ölçüm düzeyinde de klasik CNN’e ve D-CNN’e göre daha yüksek başarımlar elde etmiştir. Klasik CNN modelinin en iyi durumda elde ettiği başarı oranı %89,97 iken, OptiConNet modelinde bu değer %92,55’e yükseltmiştir. Benzer şekilde ortalama doğrulukta klasik CNN %89,65 olarak ölçülürken, OptiConNet %92,25 ile ortalama doğruluk değerinde de klasik CNN’den daha yüksek bir başarımlar göstermiştir. Ortama ölçülen değer önerilen modelin yalnızca belirli bir koşulda değil, genel olarak daha tutarlı bir şekilde daha iyi bir sınıflandırma başarısı elde ettiğini göstermektedir. Özellikle OptiConNet’in en kötü durumda dahi %91,85’lik bir doğruluk göstermesi modelin kararlı bir şekilde klasik

CNN'den daha başarılı bir başarımlı elde ettiğini göstermektedir. Bu sonuçlar OptiConNet'in veri kümesindeki örüntüleri daha derin ve anlamlı şekilde öğrendiğini ve sonuç olarak klasik CNN'den daha iyi bir sınıflama yaptığını ortaya koymaktadır.

F1 Skoru, kesinlik ve duyarlılık değerleri birlikte değerlendirildiğinde OptiConNet modelinin klasik CNN'e göre yüksek başarımlı sergilediği görülmektedir. F1 Skoru açısından bakıldığında, klasik CNN modelinin en iyi durumda 0,900 değeri elde ettiği, buna karşın OptiConNet'in 0,929'a kadar ulaştığı görülmektedir. Bu fark önerilen modelin hem doğru pozitif oranını artırdığını hem de yanlış sınıflandırmaları azalttığını göstermektedir. Kesinlik değerleri incelendiğinde CNN'in ortalama 0,875 düzeyinde kalırken. OptiConNet'in 0,922'ye ulaşması, önerilen modelde yanlış pozitifleri azalttığını ortaya koymaktadır. Benzer şekilde duyarlılık açısından da OptiConNet'in 0,920 civarındaki ortalama değeri, CNN'in 0,899'luk ortalama duyarlılık değerine kıyasla daha yüksek olup, modelin pozitif örnekleri daha doğru bir şekilde tespit ettiğini göstermektedir. Tüm bu metrikler birlikte değerlendirildiğinde. OptiConNet modelinin hem doğruluk hem de diğer metrikler açısından klasik CNN'e göre daha etkili ve güvenilir bir başarımlı elde ettiği görülmektedir. OptiConNet modelinin sınıflar üzerindeki sonuçlarını daha detaylı incelemek için karmaşıklık matrisi oluşturulup Şekil 3.8'de gösterilmiştir.



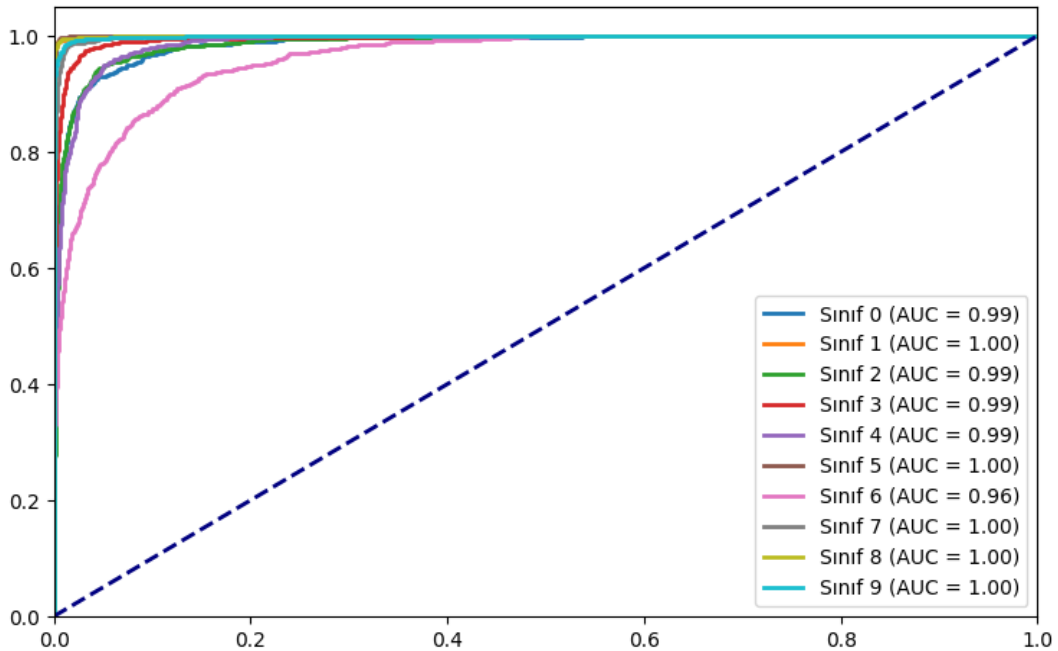
Şekil 3.8 OptiConNet modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi

Şekil 3.8’de sunulan karmaşıklık matrisi incelendiğinde, köşegen üzerindeki doğru pozitif değerlerin 1 (pantolon), 5 (sandalet), 7 (spor ayakkabı), 8 (çanta) ve 9 (bilek boy bot) sınıflarında diğer sınıflara göre daha yüksek yoğunluğa sahip olduğu görülmektedir. Bu sınıflarda modelin daha yüksek doğrulukla tahmin yapıp diğer sınıflardan doğruluğu daha yüksek olmuştur. En yüksek doğruluğun 8 (çanta) sınıfında elde edildiği görülmektedir. Bu sınıfta model %98,9’luk bir doğruluk göstererek, 1000 örnekten 989’unu doğru tahmin etmiş, sadece 11 örnekte hata yapmıştır. Hatalı tahminlerinin dağılımı sınıf başına 0-3 adet aralığında olmuştur. Bu dağılım 8 (çanta) sınıfını belirli bir başka sınıfa benzetmediğini göstermektedir.

En düşük doğruluğun 6 (gömlek) sınıfında gerçekleştiği görülmektedir. Model, bu sınıfa ait örneklerin yalnızca %72,8’ini doğru tahmin etmiş, geri kalan kısmını özellikle 0 (t-shirt) sınıfı ile sıklıkla karıştırmıştır. Gömlek sınıfına ait 120 görsel 0 (t-shirt) olarak tahmin edilirken, 0 (t-shirt) sınıfına ait 69 görsel de 6 (gömlek) sınıfı olarak hatalı tahmin yapılmıştır. Bu iki değerde sırasıyla karmaşıklık matrisinde en çok hatalı tahmin yapılan birinci ve ikinci hücreyi

gösteren değerler olmuştur. Bu modelin en çok gömlek ile tshirt sınıflarının karıştırıldığını göstermektedir. Ayrıca 6 (gömlek) sınıfına 120 kez 8 (tshirt), 60 kez 4 (ceket), 54 kez 2 (kazak), 9 kez (çanta) ve 3 kez 1 (pantolon) tahmini yapılarak toplamda 272 kez yanlış tahminde bulunulmuştur. Bu durum modelin en çok gömlek sınıfının özelliklerinin diğer giyim kategorileriyle yüksek benzerlik taşıdığını ve modelin bu sınıfta ayırt ediciliğinin düşük olduğunu göstermektedir. Ayrıca karmaşıklık matrisinin genel dağılımına bakıldığında, 6 (gömlek) sınıfının diğer sınıflar için de sıkça hatalı tahmin edilen bir kategori olduğu görülmektedir.

Pantolon sınıfında ise hem yüksek doğruluk hem de yüksek kesinlik değerine sahiptir. Model pantolon olarak sınıflandırdığı örneklerde %98,2 doğru tahminde bulunulmuştur. Ayrıca diğer sınıflara sadece 14 kez pantolon demiştir. Bu durum modelin “pantolon” dediği bir görüntünün gerçekten pantolon olma olasılığının çok yüksek olduğunu göstermektedir. Karmaşıklık matrisi yalnızca genel doğruluk oranlarını yanında modelin hangi sınıflarda daha emin veya daha kararsız davrandığını da göstermektedir. Sınıfların ayırt ediciliğini ortaya koymak için AUC –ROC grafiği Şekil 3.9’da sunulmaktadır.



Şekil 3.9 OptiConNet modeli MNIST Fashion veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.9’da sunulan AUC-ROC eğrisi, modelin her bir sınıf için gösterdiği ayırt etme

başarımını ortaya koymaktadır. Eğriler incelendiğinde, tüm sınıfların AUC değerlerinin oldukça yüksek olduğu, özellikle 1 (pantolon), 5(sandalet), 7(spor ayakkabı), 8(çanta) ve 9 (bilek bot) sınıfları için AUC değerinin 1,00 olduğu görülmektedir. Bu durum modelin bu sınıfları yüksek oranda ayırt edebildiğini göstermektedir. Modelin en düşük ayır ediciliği 6 (gömlek) sınıfı için olduğu grafikteki en aşağıdan seyir eden pembe çizgiden görülmektedir. Sonuç olarak modelin ayırt etme yeteneğinin güçlü ve sınıflandırma doğruluğunun istikrarlı olduğunu göstermekte, önerilen yöntemin sınıflar arasında etkili bir şekilde ayırım yapabildiğini desteklemektedir. Tablo 3.8’de elde edilen sonuçlar literatürdeki benzer çalışmalar ile karşılaştırmalı olarak olarak verilmiştir.

Tablo 3.8 OptiConNet modelinin MNIST Fashion veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.

Yazar / Yıl	Model	Doğruluk
(Teow 2020)	Minimal CNN	%90,16
(Sumera vd. 2022)	CNN With ADAM	%91,00
(Venkataravanappa vd. 2024b)	VGG19	%91,92
	Resnet50	%90,25
(Saranya ve Geetha 2024)	MCLO+CNN Adagrad	%91,02
	MCLO+CNN RMSProp	%89,06
(Wu vd. 2024)	HOG	%90,19
(Jian vd. 2024)	CNN	%87,48
(Absur vd. 2024)	Next-Generation CNN	%91,39
(Fakhouri vd. 2024)	SHIOGT	%92,00
(Saranirad vd. 2024)	CDNA-SNN	%90,12
(Isong 2025)	Lightweight CNN	%89,00
Önerilen Model	OptiConNet	%92,55

Tablo 3.8’de incelendiğinde 2020–2025 yılları arasında alınan örnek literatür çalışmalarında doğrulukların genellikle %89–%92 aralığında seyrettiği görülmektedir. Tablodaki çalışmalardan 2020 yılında Teow’un Minimal CNN modeli %90,16 doğruluk elde ederken, 2024 yılına kadar yapılan çalışmalarla bu oran kademeli olarak artmıştır ve özellikle Fakhouri ve arkadaşları tarafından önerilen SHIOGT modeli %92.00 ile en yüksek değere ulaşmıştır. Buna karşın, 2025 yılında geliştirilen OptiConNet modeli %92,55 doğruluk oranı ile örnek alınan literatür çalışmalarını geride bırakarak daha iyi bir başarıyı göstermiştir. Bu sonuç modelin yapısal derinliği ve optimizasyon stratejisindeki iyileştirmelerin öğrenme yeteneğini artırdığını, dolayısıyla OptiConNet’in literatürde yer alan yaklaşımlara göre iyi bir doğruluk ve genelleme kabiliyeti sunan güçlü bir alternatif olduğunu ortaya koymaktadır.

3.2.2. OptiConNet modeli Brain Tumor MRI veri kümesi sonuçları

Önerilen OptiCoNet modeli çalışmanın ilk bölümünde kıyaslama veri kümesi olarak görülen MNIST Fashion üzerinde eğitilip test edilmiştir. Bu bölümde ise gerçek hayat problemlerinden beyin tümörü hastalarından alınan manyetik rezonans görüntülerinden oluşan Brain Tumor MRI veri kümesi kullanılarak test edilmiştir. Uygulama yapılırken Brain Tumor MRI veri kümesi için üç tane evrişim katmanı kullanılmıştır. Birinci evrişim katmanında 32 adet 3×3 lük filtre kullanılmıştır. Dolgulama ve adım 1 olarak alınıp maksimum havuzlama yapılmıştır. İkinci evrişim katmanında ise 3×3 ’lük 64 filtre, adım ve dolgulama 1 olarak belirlenmiştir. Bu katmanında devamında maksimum havuzlama kullanılmıştır. Üçüncü evrişim katmanında ise 3×3 ’lük 128 filtre, adım ve dolgulama 1 olarak belirlenmiştir. Bu yapıda düzleştirme katmanında oluşan vektörler 32768×1 boyutunda oluşacaktır. Tam bağlı katmanda ise iki katman bulunmaktadır. Giriş katmanında 512 çıkış katmanında ise 4 nöron bulunan her katmanda ReLU aktivasyon fonksiyonu kullanılmıştır. Modelin içerisinde yer alan SCO algoritması için; maksimum iterasyon 100, maksimum başarısız güncelleme eşiği 5, b parametresi 2,4, arama uzayının minimum ve maksimum sınırları 0-5712 olarak belirlenmiştir. Bu parametreler ile yapılan deneysel uygulamada elde edilen sonuçlar, her bir modelin beş kez ayrı ayrı eğitilmesi ve bu eğitimlerden elde edilen test çıktılarının değerlendirilmesiyle oluşturulmuştur. Ayrıca önerilen OptiConNet modeli ile aynı parametrelere sahip klasik CNN oluşturulup elde edilen sonuçlar karşılaştırılmıştır. Uygulamada beş kez yapılan tekrarlı eğitim ve test süreçleri neticesinde elde edilen karşılaştırma tablosu Tablo 3.9’da verilmiştir.

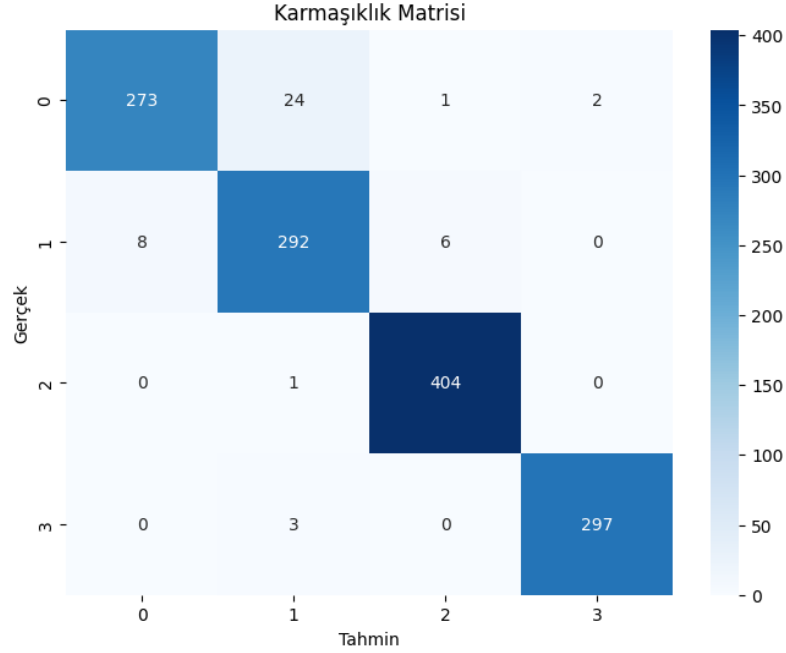
Tablo 3.9 OptiConNet modeli Brain Tumor MRI veri kümesi test sonuçlarının karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	90,82	0,909	0,908	0,907
	Ortalama	91,22	0,911	0,911	0,912
	En İyi	91,94	0,919	0,919	0,920
OptiConNet	En Kötü	95,95	0,955	0,959	0,958
	Ortalama	96,34	0,963	0,963	0,964
	En İyi	96,56	0,968	0,964	0,966

Tablo 3.9'a bakıldığında klasik CNN modeli ile önerilen OptiConNet modeli arasındaki doğruluk farkı oldukça belirgindir. CNN modeli en iyi başarımda %91,94 doğruluk elde ederken, OptiConNet modeli %96,56'ya ulaşarak doğrulukta %4,6'lık bir artış göstermiştir. Ortalama doğrulukta da benzer bir fark görülmektedir; klasik CNN %91,22 iken, OptiConNet %96,34 ile daha yüksek başarıyı sergilemektedir. Ortalama değerler arasındaki oluşan bu fark modelin kararlı bir şekilde klasik CNN'den daha başarılı olduğunu göstermektedir. En kötü durumda dahi OptiConNet %95,95 ile oldukça yüksek bir doğruluk sağlamakta, en kötü sonucu veren modelin bile daha iyi başarımlar gösterdiğini göstermektedir. Bu sonuçlar ile OptiConNet'in doğruluk açısından CNN'e kıyasla belirgin bir üstünlük sağladığı ve daha etkili bir sınıflandırma yapabildiği söylenebilir.

F1 Skoru, kesinlik ve duyarlılık değerleri açısından tablo incelendiğinde, OptiConNet modeli CNN'e kıyasla tüm metriklerde daha yüksek ve dengeli başarımlar göstermektedir. F1 Skorunda CNN'in en iyi değeri 0,919 iken, OptiConNet 0,968 ile çok daha yüksek bir dengede doğruluk ve duyarlılık sunmaktadır. Bu modelin hem doğru pozitifleri artırdığını hem de yanlış sınıflandırmaları minimize ettiğini göstermektedir. Kesinlik değerleri incelendiğinde, OptiConNet ortalama 0,963 değerine ulaşarak CNN'in 0,911 değerine kıyasla yanlış pozitifleri önemli ölçüde azaltmıştır. Duyarlılık açısından da OptiConNet'in ortalama 0,964 değeri CNN'in 0,912 değerinin oldukça üzerinde olup, modelin pozitif örnekleri tespit etme yeteneğinin arttığını göstermektedir. Tüm bu metrikler birlikte değerlendirildiğinde, OptiConNet modelinin yalnızca daha yüksek doğruluk değil, aynı zamanda daha dengeli ve

güvenilir bir sınıflandırma başarımı sergilediği görülmektedir. Modelin başarımının sınıf bazında değerlendirilebilmesi amacıyla karmaşıklık matrisi oluşturulmuş ve Şekil 3.10'da sunulmuştur.



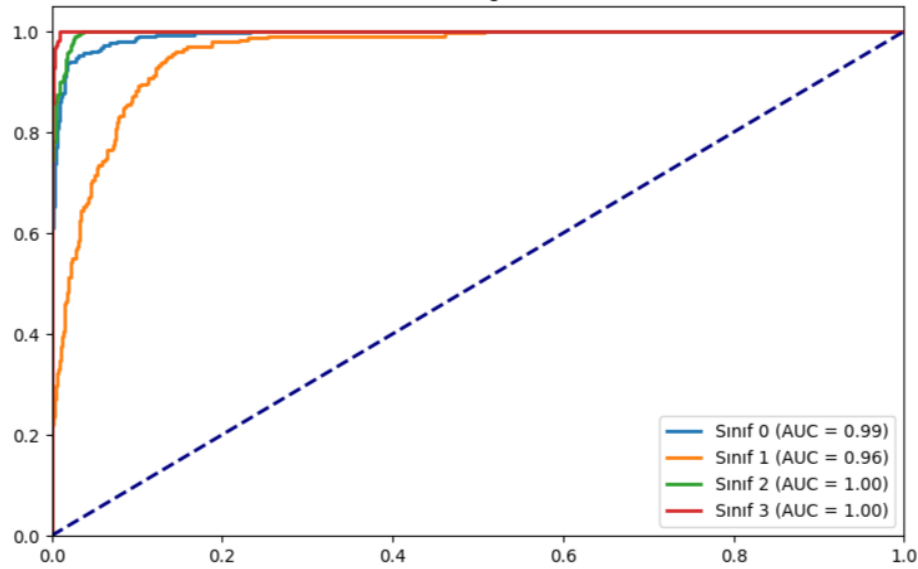
Şekil 3.10 OptiConNet modelinin Brain Tumor MRI veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.

Şekil 3.10'da sunulan karmaşıklık matrisi incelendiğinde modelin dört farklı beyin tümörü sınıfı olan 0 (Glioma), 1 (Meningioma), 2 (No Tumor) ve 3 (Pituitary) kategorilerinde genel olarak yüksek doğrulukla sınıflandırma yaptığı görülmektedir. Köşegen üzerindeki doğru pozitif değerler incelendiğinde, özellikle 2 (No Tumor) sınıfında modelin çok güçlü bir başarımla sergilediği ve %99,9'luk bir doğruluk ile neredeyse tüm örnekleri doğru sınıflandırdığı anlaşılmaktadır. Model bu sınıfta yer alan 405 örneğin 404'ünü doğru tahmin etmiş, yalnızca 1 örnekte hata yapmıştır. Bu sonuç modelin tümör bulunmayan beyin MR görüntülerini diğer tümör tiplerinden ayırt etmede oldukça başarılı olduğunu göstermektedir.

Modelin benzer biçimde yüksek başarı gösterdiği bir diğer sınıf 3 (Pituitary) sınıfıdır. Bu sınıfta model 300'e yakın örneğin 297'sini doğru tahmin etmiş, yalnızca 3 tanesini yanlış biçimde 1 (Meningioma) olarak sınıflandırmıştır. Bu durum pituitary tümörlerin diğer tümörlere göre model tarafından daha belirgin ve ayırt edilebilir olduğunu göstermektedir. Model bu iki sınıfın gerisinde kalsa da 0 (Glioma) ve 1 (Meningioma) sınıflarını da yüksek oranda doğru tahmin etmiştir. 0 (Glioma) sınıfında model 273 örneği doğru tahmin ederken 24

örneği 1 (Meningioma) olarak karıştırmıştır. Bu karışıklığın nedeni, glioma ve meningioma tümörlerinin bazı MRI görsellerinde benzerlikler olması olarak düşünülebilir. 1 (Meningioma) sınıfında ise 292 doğru tahmin yapılmış 8 örnek 0 (Glioma), 6 örnek ise 2 (No Tumor) sınıfına hatalı olarak tahmin edilmiştir.

Genel olarak değerlendirildiğinde modelin No Tumor ve Pituitary sınıflarında hem doğruluk hem de kesinlik açısından yüksek başarımla sergilediği Glioma ve Meningioma sınıflarında ise bazı çapraz karışıklıkların yaşandığı görülmektedir. Karmaşıklık matrisi modelin özellikle tümör tespiti gereken sınıflarda güçlü bir ayırt edicilik yeteneğine sahip olduğunu ve genel başarımın yüksek olduğunu ortaya koymaktadır. Yapılan incelemelerden sonra sınıfların ayırt ediciliğini görselleştirmek için AUC-ROC grafiği çizdirilip Şekil 3.11’de sunulmuştur.



Şekil 3.11 OptiConNet modeli Brain Tumor MRI veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.11’de verilen AUC-ROC eğrisi, modelin dört farklı sınıf için ayırt etme başarımını göstermektedir. Grafikte yer alan eğriler incelendiğinde. 2 (no tumor) ve 3 (pituitary) sınıfları için AUC değerlerinin 1,00 olduğu, dolayısıyla modelin bu sınıfları doğru biçimde ayırt ettiği görülmektedir. 0 (glioma) sınıfı için AUC değeri 0,99 olup modelin bu sınıfa ilişkin oldukça yüksek bir sınıflandırma doğruluğuna sahip olduğunu göstermektedir. Ayırt ediciliği en düşük sınıf ise 0,96 AUC değeri ile 1 (Meningioma) sınıfı olmuştur. Genel olarak tüm sınıflarda yüksek AUC değerlerinin elde edilmesi modelin sınıflar arasında başarılı bir ayırım gerçekleştirdiğini göstermektedir. Modelin yüksek doğruluk elde ettiği başarımı ayrıca literatürde yapılan çalışmalar ile de karşılaştırılıp, Tablo 3.10’da verilmiştir.

Tablo 3.10 OptiConNet modelinin Brain Tumor MRI veri kümesi için literatürdeki çalışmalar ile karşılaştırılması.

Çalışma	Model	Doğruluk
(Vankdothu ve Hameed 2022)	Backpropagation	%88,83
	U-Net	%90,86
	Proposed RCNN	%95,17
(Özkaraca vd. 2023)	VGG16	%91,03
	DenseNET	%90,30
	Modified CNN	%96,00
(Deepak ve Ameer 2023)	CNN (Lce. Lwf) + SVM	%95,40
	CNN (Lwf) + SVM	%94,60
(Rahman ve Islam 2023)	CNN	%88,24
	PDCNN	%96,00
(Gómez-Guzmán vd. 2023)	EfficientNetB0	%90,88
	Xception	%95,67
(Shamshad vd. 2024)	MobileNet	%87,00
(Khan vd. 2024)	Hybrid-NET	%95,10
	DensNet169	%88,00
	DenseNet169 + SVM	%92,00
Önerilen Model	OptiConNet	%96,56

Tablo 3.10 incelendiğinde, 2020–2025 yılları arasında yapılan örnek literatür çalışmalarında doğruluk metriğinin genellikle %88 ile %96 aralığında seyrettiği görülmektedir. Tablodaki literatür örnekleri arasında yer alan Özkaraca ve arkadaşlarının çalışması ile Rahman ve Islam tarafından önerilen modeller en yüksek düzeyde %96,00 doğruluk bildirmiştir. Buna karşın, önerilen OptiConNet modeli %96,56 doğruluk oranına ulaşarak bu çalışmalarda rapor edilen değerleri aşmış ve literatürdeki mevcut yaklaşımlara kıyasla daha üstün bir başarıyı

ortaya koymuřtur. Bu durum modelin katman yapısında yapılan optimizasyonların ve karma özellik ıkarım yönteminin sınıflandırma dođruluđunu artırdıđını gstermektedir. OptiConNet'in dođruluđu hem genel amalı veri kmelerinde hem de tıbbi grnt analizi gibi uzmanlık gerektiren bir alanda yksek dođruluk ve kararlılık ile dođrulanmıřtır, bu da modelin ok amalı ve gvenilir bir model olduđunu ortaya koymaktadır.

3.3. HyBee-CNN Modeli MNIST Fashion Veri Kmesi Sonuları

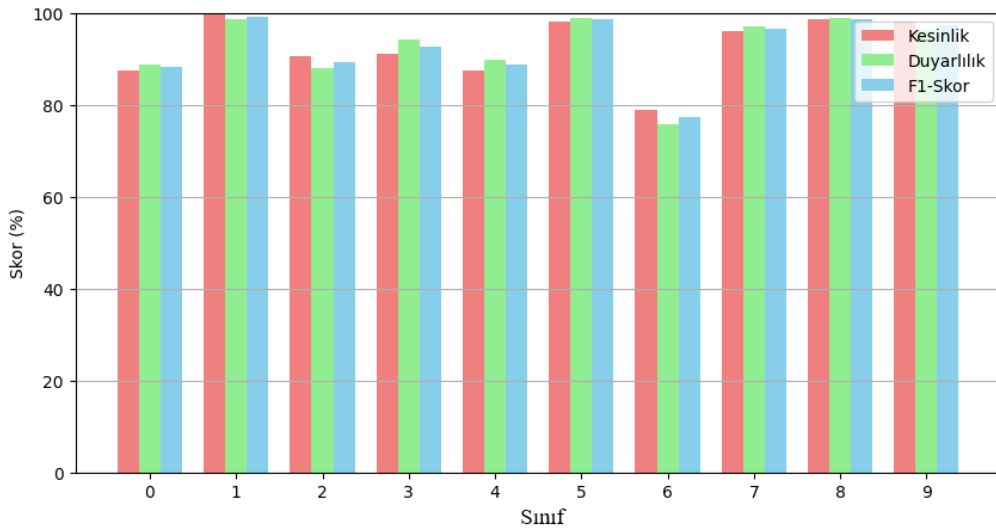
HyBee-CNN modeli diđer nerilen modellerde de kullanılan MNIST Fashion veri kmesi zerinde eđitilip test edilerek, test sonuları incelenmiřtir. HyBee-CNN modelinde ilk olarak evriřim katmanında 3×3'lk 32 filtre ve 1 dolgulama, ikinci evriřim katmanında 3×3'lk 32 filtre ve 1 dolgulama ve 0,2 ayrılma oranı (dropout) bulunmaktadır. Tam bađlı katmanda, birinci katman 128, ikinci katman 10 nrondan oluřmaktadır. Kısıtlı ađrılık reticinin kullandıđı ABC algoritmasının bařlangı parametreleri 50 arı, gncellenmeme limiti 10 ve iterasyon sayısı ise 100 olarak belirlenmiřtir. Belirlenen bu parametreler ile alıřtırılan HyBee-CNN ile OptiConNet, D-CNN ve klasik CNN modellerinin dođruluk, duyarlılık, kesinlik ve F1 Skor metrikleri Tablo 3.11'de verilmiřtir.

Tablo 3.11 HyBee-CNN modeli MNIST Fashion veri kümesi test sonuçlarının karşılaştırılması.

Model	Başarım	Doğruluk	F1 Skoru	Kesinlik	Duyarlılık
CNN	En Kötü	88,91	0,881	0,890	0,867
	Ortalama	89,65	0,899	0,875	0,899
	En İyi	89,97	0,900	0,901	0,899
D-CNN	En Kötü	91,51	0,916	0,916	0,915
	Ortalama	91,80	0,918	0,917	0,917
	En İyi	92,11	0,921	0,922	0,921
OptiConNet	En Kötü	91,85	0,918	0,919	0,915
	Ortalama	92,25	0,928	0,922	0,920
	En İyi	92,55	0,929	0,926	0,922
HyBee-CNN	En Kötü	92,20	0,922	0,922	0,920
	Ortalama	92,46	0,928	0,927	0,925
	En İyi	92,75	0,929	0,928	0,927

Tablo 3.11’de verilen başarımlar değerlendirildiğinde HyBee-CNN’in diğer önerilen modellerden ve klasik CNN modelinden en kötü, ortalama ve en iyi olmak üzere tüm ölçüm düzeylerinde daha yüksek doğruluk değerleri elde ettiği görülmektedir. Özellikle HyBee-CNN gibi karma bir model olan OptiConNet modelinden %0,21 doğruluk farkı ile daha iyi doğruluk ortalaması elde etmiştir. Klasik CNN modelinin en iyi durumda ulaştığı doğruluk oranı %89,97 seviyesinde kalırken, HyBee-CNN modelinde bu değer %92,75’e yükselmesi önerilen yaklaşımın sınıflandırma başarımını belirgin şekilde artırdığını göstermektedir. Benzer şekilde ortalama doğruluk değeri klasik CNN için %89,65 olarak ölçülmüşken, HyBee-CNN’in %92,46 ile bu değer oldukça üzerinde bir başarı sergilediği görülmektedir. Ortalama başarımdaki bu artış genel olarak daha tutarlı ve daha yüksek sınıflandırma başarısı elde ettiğini ortaya koymaktadır. Özellikle HyBee-CNN’in en kötü durumda dahi %92,20 doğruluk oranına ulaşması, modelin kararlı yapısını ve veri kümesindeki örüntüleri daha etkin şekilde temsil edebildiğini göstermektedir.

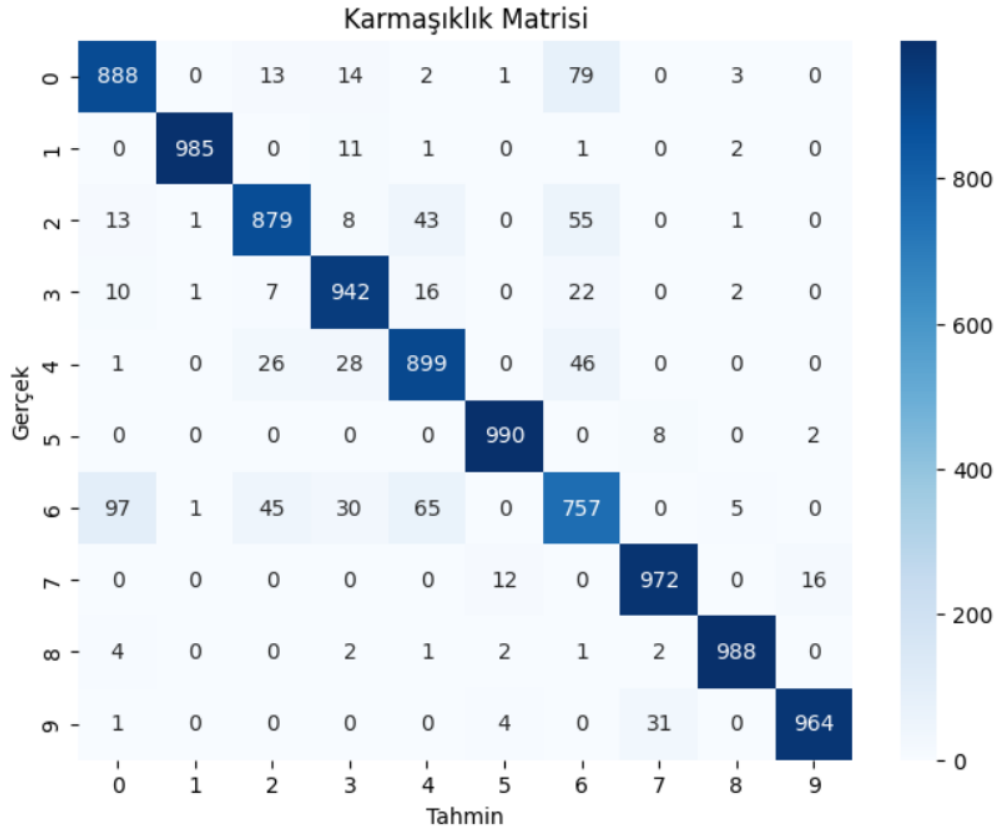
F1 Skoru, kesinlik ve duyarlılık metrikleri birlikte değerlendirildiğinde HyBee-CNN modelinin klasik CNN'e göre her üç ölçüm düzeyinde de daha yüksek başarımlar sergilediği görülmektedir. F1 Skoru açısından bakıldığında, klasik CNN modelinin en iyi durumda elde ettiği 0,900 değerine karşın HyBee-CNN'in 0,929 seviyesine kadar ulaştığı görülmektedir. Bu fark, önerilen modelin hem doğru pozitif oranını artırdığını hem de yanlış sınıflandırmaları azalttığını göstermektedir. Kesinlik değerleri incelendiğinde, klasik CNN modelinin ortalama 0,875 düzeyinde kaldığı, buna karşın HyBee-CNN modelinin ortalama 0,927 kesinlik elde ettiği görülmektedir. Bu durum önerilen modelin yanlış pozitif üretme olasılığını önemli ölçüde azalttığını ve daha güvenilir bir sınıflandırma gerçekleştirdiğini göstermektedir. Duyarlılık açısından da benzer bir eğilim gözlemlenmekte olup, HyBee-CNN'in ortalama 0,925 duyarlılık değeri, klasik CNN'in 0,899 seviyesindeki duyarlılık değerine kıyasla daha yüksek olup modelin pozitif örnekleri daha doğru bir şekilde tespit ettiğini ortaya koymaktadır. Tüm bu metrikler birlikte değerlendirildiğinde, HyBee-CNN modelinin doğruluk, F1 Skoru, kesinlik ve duyarlılık açısından klasik CNN'e göre daha etkili ve daha güvenilir bir başarımlar ortaya koyduğu görülmektedir. F1 Skoru, kesinlik ve duyarlılık metriklerinin her bir sınıf için dağılımı Şekil 3.12'de verilmiştir.



Şekil 3.12. HyBee-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait sınıf bazlı kesinlik, duyarlılık ve F1 Skoru dağılımı.

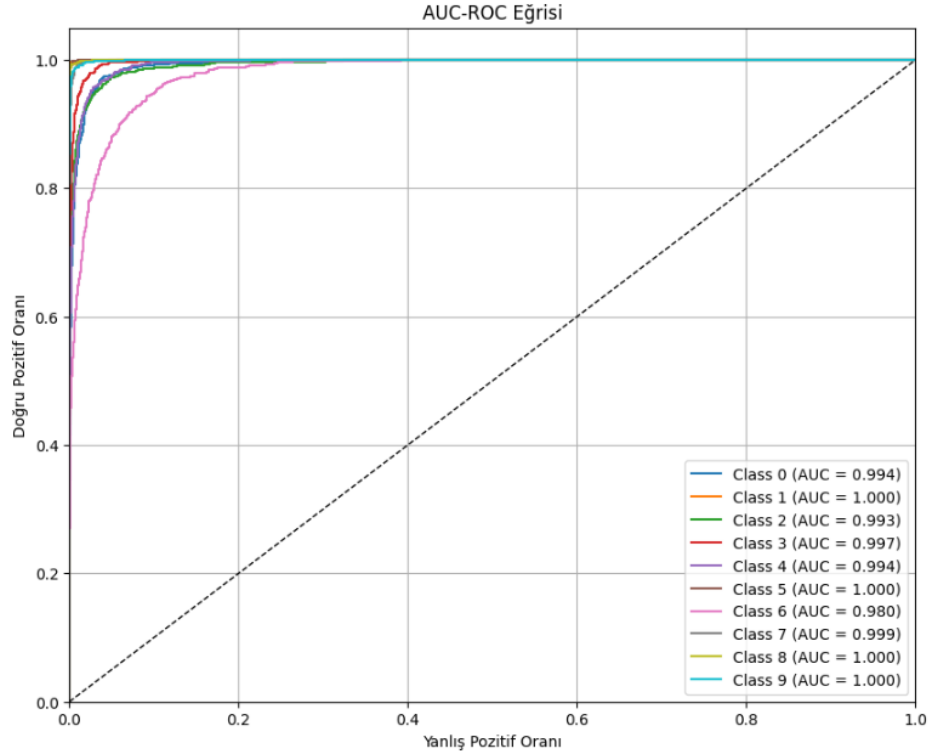
Şekil 3.12'de verilen dağılıma bakıldığında model en düşük başarımları 6 (Gömlek) sınıfı için yapıırken, 0 (Tişört), 2 (Kazak), 3 (Elbise), 4 (Kaban) sınıfları diğer sınıflara göre daha düşük başarımlar göstermiştir. HyBee-CNN modeli 1 (Pantolon), 5 (Sandalet), 7 (Spor Ayakkabı), 8 (Çanta), 9 (Bot) sınıfları için kusursuza yakın başarımlar göstermiştir. Bu dağılımın detaylarının

incelenebileceği karmaşıklık matrisi Şekil 3.13'te verilmiştir.



Şekil 3.13. HyBee-CNN modelinin MNIST Fashion veri kümesi üzerindeki test sonuçlarına ait karmaşıklık matrisi.

Şekil 3.13'te verilen karmaşıklık matrisinde en başarılı sınıflar 1 (Pantolon), 5 (Sandalet), 8 (Çanta) olmuştur. Özellikle bu sınıfların, diğer sınıflarda yanlış olarak tahmin edilme oranları çok düşüktür. Bu durum 1 (Pantolon), 5 (Sandalet), 8 (Çanta) sınıflarının kesinlik ve duyarlık metriklerinde de yüksek başarımlar göstermesini sağlamaktadır. Model en kötü başarımını 6 (Gömlek) sınıfı tahmin ederken göstermiştir. Ayrıca 6 (Gömlek) sınıfı özellikle 0 (Tişört), 2 (Kazak) ve 4 (Kaban) hatalı olarak tahmin edilmiştir. Bu durum sadece 6 (Gömlek) sınıfını bilmediğini değil, 0 (Tişört), 2 (Kazak), 4 (Kaban) sınıflarını tahmin ederken o sınıflara da 6 (Gömlek) diyerek o sınıfların tahmin edilmesini zorlaştırmaktadır. Bu durumun tam tersi birbirine hiç karıştırılmayan sınıflar 8 (Çanta) ve 9 sınıflarında, 4 (Kaban) ve 5 (Sandalet) sınıflarında, 0 (Tişört) ve 1 (Pantolon) sınıflarında olmuştur. Bu verilen ikili sınıflar birbirlerine hiç karıştırılmamış ve birbirleri arasındaki ayırt edicilikleri kusursuz olduğu görülmüştür. Modelin genel olarak sınıfların ayırt ediciliği görselleştirmek için çizilen AUC-ROC grafiği Şekil 3.14'te verilmiştir.



Şekil 3.14 HyBee-CNN modeli Brain Tumor MRI veri kümesi test sonuçları AUC-ROC grafiği.

Şekil 3.14’te verilen AUC-ROC eğrisi incelendiğinde modelin tüm sınıflarda oldukça yüksek ayırt etme başarımı gösterdiği görülmektedir. Birçok sınıfta AUC değerinin 0.99 ve üzeri olması, modelin pozitif ve negatif örnekleri neredeyse kusursuz şekilde ayırabildiğini göstermektedir. Sınıf 6’nın (Gömlek) ayırt ediciliği diğer sınıflara kıyasla daha düşük olduğu görülmektedir. HyBee-CNN modelinin genel olarak sınıflar üzerinde dengeli ve güvenilir bir ayırım yapabildiğini açıkça ortaya koymaktadır. HyBee-CNN modeli incelenen bu metriklerde yüksek başarımla eğitim süresini uzattığı görülmüştür.

HyBee-CNN modelinin ve tez çalışmasında önerilen diğer modellerin MNIST Fashion veri kümesinde eğitim süreleri, klasik CNN modeliyle yapılan beş eğitim süresinin ortalaması saniye cinsinden karşılaştırılması Tablo 3.12’de sunulmuştur.

Tablo 3.11 Önerilen modellerin çalışma sürelerinin saniye cinsinden karşılaştırılması

	CNN	D-CNN	C-CNN	S-CNN	RS-CNN	M-CNN	OptiConNet	HyBee-CNN
Süre (Sn.)	834	973	945	942	953	979	1050	1127

Tablo 3.12’de verilen eğitim sürelerini çalışan sezgisel algoritmaların başlangıç parametreleri, veri kümelerindeki görsellerin boyutları ve kullanılan GPU modelleri süreleri doğrudan etkilemektedir. Bu tabloda verilen aynı veri kümesinde, aynı GPU üzerinde ve aynı eğitim tur (epoch) ile ölçülmüş değerlerdir. En yavaş çalışan HyBee-CNN olurken, En hızlı çalışan Klasik CNN yöntemi olmuştur. HyBee-CNN’in 293 saniye bir zaman kaybına sebep olduğu görülmektedir. Bu kayıp HyBee-CNN’nin parametrelerini düzenleyerek minimize edilebilir.

4. SONUÇLAR

Bu çalışma kapsamında, derin öğrenme modellerinin eğitiminde rastgele ağırlık belirlenmesinden kaynaklanan öğrenme kararsızlıkları, yavaş yakınsama ve yerel minimumlara takılma gibi temel zorluklara karşı iki aşamalı özgün bir çözüm önerilmiştir. Çalışma klasik CNN yapısını geliştirip, kısıtlı ağırlık üretim yöntemi ile eğitim sürecinin optimizasyonunu hedeflemiştir. Bu kapsamda önerilen modeller, farklı veri kümeleri üzerinde kapsamlı biçimde test edilerek hem klasik CNN hem de literatürdeki güncel çalışmalarla karşılaştırılmıştır.

Tez çalışmanın ilk aşamasında, kısıtlı CNN mimarisi çatısı altında beş farklı model C-CNN, D-CNN, S-CNN, M-CNN ve RS-CNN önerilmiştir. Bu modellerde, tam bağlı katmandaki ağırlıkların rastgele değil, belirli matematiksel kısıtlar altında üretilmesi sağlanmıştır ve böylece ağırlıkların eğitime daha uygun ağırlıklar ile başlanmıştır. Veri kümesi ilk olarak CNN'in özellik çıkarım katmanından geçirilip özellik vektörleri elde edilmiştir. Elde edilen özellik vektörlerinden önerilen modelin algoritmasına uygun şekilde vektör seçimi yapılarak, seçilen vektörlerden fark ya da toplam vektörleri oluşturulup bağlantı ağırlıklarının başlangıç parametreleri üretilmiştir. Üretilen bu ağırlıklar ile model eğitilip CNN'in doğruluğu artırılmıştır. Önerilen bu beş modeli deneysel olarak incelemek için literatürde sıklıkla kullanılan MNIST, MNIST Fashion ve CIFAR-10 veri kümelerinde eğitilip test edilmiştir.

Deneysel sonuçlar göstermiştir ki, önerilen bu kısıtlı modeller klasik CNN yapısına kıyasla daha yüksek doğruluk, F1 Skoru, kesinlik ve duyarlılık değerleri elde etmiştir. Özellikle D-CNN ve C-CNN modelleri kullanılan üç veri kümesi içinde en iyi doğruluk gösteren iki model varyantı olmuştur. Bu dikkat çekici sonuç her iki yöntemin temelinde yer alan fark vektörünün kullanımından kaynaklanmaktadır. Kısıtlı yöntemler incelendiğinde CNN ile oluşturulan karma yöntemlerde toplam vektörü temelli modeller, fark vektörü temelli modellerin gerisinde kalmıştır. Ayrıca önerilen modellerin en iyi sonucu ile klasik CNN yöntemi arasında oluşan fark MNIST, MNIST Fashion ve CIFAR-10 veri kümeleri bazında incelendiğinde sırasıyla %0,6, %1,56 ve %5,67 olmuştur. Sıralamaya dikkat edildiğinde veri kümesindeki problem zorlaştıkça önerilen modelin klasik CNN'e karşı elde ettiği yüzdelik doğruluk farkı artmaktadır. Bu bulguya dayanarak önerilen modellerin problem zorlaştıkça daha fazla fayda sağladığı görülmektedir. Ayrıca MNIST veri kümesinde sadece önerilen iki model klasik CNN'den daha iyi başarımlar gösterirken, CIFAR-10 veri kümesinde önerilen beş modelin tamamı klasik CNN'den daha iyi başarımlar göstermiştir. Alınan sonuçlar problem karmaşıklığı arttıkça modelin daha etkili olduğu göstermektedir. Çalışmanın bu bölümünde

elde edilen fark vektörü temelli modellerin başarımlarında daha iyi sonuçlar elde etmesi, çalışmanın devamında sezgisel algoritmalar ile yapılacak olan karma yapıya da ışık tutmuştur.

Tez çalışmasının ikinci aşamasında, birinci çalışmada önerilen kısıtlı CNN mimarileri sezgisel bir yaklaşımla geliştirilmiş ve OptiConNet adı verilen yeni bir karma model daha önerilmiştir. OptiConNet modelinde kısıtlı ağırlık üretici, ağırlıkları hesaplarken kullanacağı vektörler SCO algoritması ile seçilmektedir. Bu sayede ağırlık parametre uzayında daha uygun bir başlangıç konumundan öğrenmeye başlaması hedeflenmiştir. OptiConNet modeli, MNIST Fashion ve gerçek hayat problemlerindeki sonuçlarını izlemek adına Brain Tumor MRI veri kümelerinde test edilmiştir. Her iki veri kümesi için de OptiConNet modeli klasik CNN modelinden daha yüksek başarımlar göstermiştir. Birinci çalışmada tespit edildiği gibi OptiConNet modelinde de veri kümesi problem karmaşıklığı arttıkça model doğruluğu artmıştır. MNIST Fashion veri kümesinde klasik CNN %89,65 doğruluk elde ederken OptiConNet modeli aynı veri kümesinde %92,55 doğruluk elde etmiştir. Birinci çalışmadaki MNIST Fashion veri kümesindeki sonuçlarda en yüksek doğruluk sağlayan D-CNN modeli %92,11 doğruluk elde etmişti, bu sonuçlara göre aynı veri kümesinde OptiConNet modeli %1,39'luk daha iyi bir sonuç göstermiştir. Bu sonuç önerilen ikinci modelin ilk modele göre daha başarılı olduğunu göstermektedir. Elde edilen deneysel sonuçlar ayrıca güncel literatürdeki diğer modeller ile karşılaştırılmış ve alınan literatür örneklemindeki çalışmalardan daha yüksek başarımlar göstermiştir. Bu sebeple önerilen OptiConNet modeli literatürde alternatif bir derin öğrenme modeli olarak gösterilebilir.

OptiConNet modelinde kullanılan tek adaylı SCO algoritmasına alternatif olarak, sürü tabanlı bir sezgisel yöntem olan ABC algoritması entegre edilerek HyBee-CNN modeli geliştirilmiştir. Bu modelin tercih edilme nedeni, ABC algoritmasının çözüm uzayını SCO'ya kıyasla farklı bir şekilde taraması ve böylece başlangıç ağırlıklarının daha etkili biçimde optimize edilmesine olanak sağlamasıdır. Önerilen model MNIST Fashion veri kümesi üzerinde eğitilip test edilmiş ve yapılan beş tekrarlı deney sonucunda HyBee-CNN, %92,75 ile en yüksek doğruluk değerine ulaşmıştır. Bu doğruluk seviyesi, yalnızca klasik CNN modelinden değil, tez kapsamında önerilen diğer altı modelden de daha iyi başarımlar göstermiştir. Bununla birlikte, doğruluk artışına rağmen modelin eğitim süresi klasik CNN'e göre yaklaşık %34 oranında daha uzun sürmüştür. Bu ek hesaplama maliyetinin azaltılması amacıyla ilerleyen çalışmalarda ABC algoritmasına ait parametrelerin iyileştirilmesi veya daha hafif sürü tabanlı alternatiflerin kullanılması değerlendirilebilir.

Sonuç olarak sezgisel algoritmaların özellikle kısıtlı CNN mimarilerinde kullanımı, klasik eğitim yöntemlerine kıyasla ağırlıkların daha dengeli dağıtılmasını ve parametre uzayının daha verimli taranmasını sağlamıştır. Böylece derin öğrenme modellerinin optimizasyon sürecine sezgisel yaklaşımların entegre edilmesi, yalnızca başarımların artışı sağlamakla kalmamış, aynı zamanda kesinlik ve kararlılık gibi metriklerde de CNN modelinin doğruluğunu artırmıştır. Bu tez kapsamında geliştirilen yaklaşımlar yalnızca görsel sınıflandırma problemleriyle sınırlı olmayıp, benzer şekilde farklı derin öğrenme alanlarında (örneğin nesne tespiti, tıbbi görüntü analizi, metin sınıflandırma vb.) uygulanabilir potansiyele sahiptir. Özellikle OptiConNet gibi hem kısıtlı hem de sezgisel optimizasyon tabanlı karma modeller, gelecekte derin öğrenme mimarilerinin eğitimi için yeni bir alternatif oluşturma potansiyeline sahiptir.

Çalışmada sürü tabanlı ABC algoritması ile tek aday tabanlı SCO algoritmaları kullanılmıştır. Literatürde optimizasyon için kullanılan birçok farklı algoritma bulunmaktadır. Gelecek çalışmalarda karma modeli oluşturmak farklı sezgisel algoritmalar kullanılabilir. Ayrıca transfer öğrenme yöntemleri için de benzer bir karma yapı oluşturulabilir.

KAYNAKÇA

- Abdel-Basset, M., L. Abdel-Fatah, ve A. K. Sangaiah. (2018). "Metaheuristic algorithms: A comprehensive review". *Computational intelligence for multimedia big data on the cloud with engineering applications* 185-231.
- Absur, M. N., K. F. A. Nasif, S. Saha, ve S. N. Nova. (2024). "Revolutionizing image recognition: Next-generation cnn architectures for handwritten digits and objects". S. 173-78 içinde *2024 IEEE Symposium on Wireless Technology & Applications (ISWTA)*. IEEE.
- Ahmed, A. A., ve S. M. Darwish. (2021). "A meta-heuristic automatic CNN architecture design approach based on ensemble learning". *IEEE Access* 9:16975-87.
- Ajit, A., K. Acharya, ve A. Samanta. (2020). "A review of convolutional neural networks". S. 1-5 içinde *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*. IEEE.
- Alzubaidi, L., J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, ve L. Farhan. (2021a). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". *Journal of big Data* 8(1):53.
- Alzubaidi, L., J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, ve L. Farhan. (2021b). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". *Journal of big Data* 8(1):53.
- Andonie, R., ve A.-C. Florea. (2020). "Weighted random search for CNN hyperparameter optimization". *arXiv preprint arXiv:2003.13300*.
- Aslam, S., ve A. B. Nassif. (2023). "Deep learning based CIFAR-10 classification". S. 1-4 içinde *2023 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE.
- Azam, B., ve N. Akhtar. (2024). "Suitability of kans for computer vision: A preliminary investigation". *arXiv preprint arXiv:2406.09087*.
- Balci, M., E. Dokur, ve U. Yuzgec. (2025). "A Hybrid LSTM-Single Candidate Optimizer Model for Short-Term Wind Power Prediction." *Computer Modeling in Engineering & Sciences (CMES)* 144(1).

- Baldominos, A., Y. Saez, ve P. Isasi. (2019). "A survey of handwritten character recognition with mnist and emnist". *Applied sciences* 9(15):3169.
- Basha, S. H. S., S. R. Dubey, V. Pulabaigari, ve S. Mukherjee. (2020). "Impact of fully connected layers on performance of convolutional neural networks for image classification". *Neurocomputing* 378:112-19.
- Bayar, B., ve M. C. Stamm. (2018). "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection". *IEEE Transactions on Information Forensics and Security* 13(11):2691-2706.
- Bbouzidi, S., G. Hcini, I. Jdey, ve F. Drira. (2024). "Convolutional neural networks and vision transformers for fashion mnist classification: A literature review". *arXiv preprint arXiv:2406.03478*.
- Belciug, S. (2022). "Learning deep neural networks' architectures using differential evolution. Case study: medical imaging processing". *Computers in biology and medicine* 146:105623.
- Bhatnagar, S., D. Ghosal, ve M. H. Kolekar. (2017). "Classification of fashion article images using convolutional neural networks". S. 1-6 içinde *2017 Fourth International Conference on Image Information Processing (ICIIP)*. IEEE.
- Cao, W., X. Wang, Z. Ming, ve J. Gao. (2018). "A review on neural networks with random weights". *Neurocomputing* 275:278-87.
- Carrington, A. M., D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, B. Wernly, C. Bennett, S. Hawken, O. Magwood, ve Y. Sheikh. (2022). "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(1):329-41.
- Chauhan, R., K. K. Ghanshala, ve R. C. Joshi. (2018). "Convolutional neural network (CNN) for image detection and recognition". S. 278-82 içinde *2018 first international conference on secure cyber computing and communication (ICSCCC)*. IEEE.
- Cheng, B., Z. Li, B. Xu, C. Dang, ve J. Deng. (2021). "Target detection in remote sensing image based on object-and-scene context constrained CNN". *IEEE Geoscience and Remote Sensing Letters* 19:1-5.
- Cheng, H. J.-H. (2020). *Empirical Study on the Effect of Zero-Padding in Text Classification*

- with CNN. University of California, Los Angeles.
- Cui, J., R. Wang, S. Si, ve C.-J. Hsieh. (2023). “Scaling up dataset distillation to imagenet-1k with constant memory”. S. 6565-90 içinde *International Conference on Machine Learning*. PMLR.
- Çetiner, H., ve S. Metlek. (2025). “ABC+ CNN-SH: Detection of peruvian coffea leaf diseases with a new hybrid classification algorithm based on ABC optimization and CNN”. *Computers and Electronics in Agriculture* 239:111114.
- Deepak, S., ve P. M. Ameer. (2023). “Brain tumor categorization from imbalanced MRI dataset using weighted loss and deep feature fusion”. *Neurocomputing* 520:94-102.
- Dogaru, R., ve I. Dogaru. (2021). “NL-CNN: a resources-constrained deep learning model based on nonlinear convolution”. S. 1-4 içinde *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*. IEEE.
- Doğan, Y. (2023). “Which pooling method is better: Max, avg, or concat (Max, Avg)”. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 66(1):95-117.
- Doon, R., T. K. Rawat, ve S. Gautam. (2018). “Cifar-10 classification using deep convolutional neural network”. S. 1-5 içinde *2018 IEEE punecon*. IEEE.
- Duan, M., Kenli Li, C. Yang, ve Keqin Li. (2018). “A hybrid deep learning CNN–ELM for age and gender classification”. *Neurocomputing* 275:448-61.
- Eltoukhy, M. M., F. S. Alsubaei, A. M. Mortda, ve K. M. Hosny. (2025). “An efficient convolution neural network method for copy-move video forgery detection”. *Alexandria Engineering Journal* 110:429-37.
- Erkoç, T., ve M. T. Eski. (2023). “A novel similarity based unsupervised technique for training convolutional filters”. *IEEE Access* 11:49393-408.
- Fahmy Amin, M. (2023). “Confusion matrix in three-class classification problems: A step-by-step tutorial”. *Journal of Engineering Research* 7(1):0.
- Fakhouri, H. N., S. Alawadi, F. M. Alwaysheh, ve F. Hamad. (2024). “Novel hybrid success history intelligent optimizer with gaussian transformation: Application in cnn hyperparameter tuning”. *Cluster Computing* 27(3):3717-39.

- Gandomi, A. H., X.-S. Yang, S. Talatahari, ve A. H. Alavi. (2013). "Metaheuristic algorithms". *Metaheuristic applications in structures and infrastructures* 1-24.
- Gholamalinezhad, H., ve H. Khosravi. (2020). "Pooling methods in deep neural networks, a review". *arXiv preprint arXiv:2009.07485*.
- Gómez-Guzmán, M. A., L. Jiménez-Beristáin, E. E. García-Guerrero, O. R. López-Bonilla, U. J. Tamayo-Perez, J. J. Esqueda-Elizondo, K. Palomino-Vizcaino, ve E. Inzunza-González. (2023). "Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks". *Electronics* 12(4):955.
- Greeshma, K. V, ve K. Sreekumar. (2019). "Fashion-MNIST classification based on HOG feature descriptor using SVM". *International Journal of Innovative Technology and Exploring Engineering* 8(5):960-62.
- Grim, J., ve P. Somol. (t.y.). "A Statistical Review of the MNIST Benchmark Data Problem".
- Grønningsæter, Y., H. S. Smørvik, ve O.-C. Granmo. (2024). "An optimized toolbox for advanced image processing with Tsetlin machine composites". S. 1-8 içinde *2024 International Symposium on the Tsetlin Machine (ISTM)*. IEEE.
- Guo, C., Y. Liu, ve X. Jiao. (2019). "Study on the influence of variable stride scale change on image recognition in CNN". *Multimedia Tools and Applications* 78(21):30027-37.
- Haider, U., M. Hanif, A. Rashid, ve S. F. Hussain. (2023). "Dictionary-enabled efficient training of ConvNets for image classification". *Image and Vision Computing* 135:104718.
- Hassen, K. B. A., J. J. M. Machado, ve J. M. R. S. Tavares. (2022). "Convolutional neural networks and heuristic methods for crowd counting: A systematic review". *Sensors* 22(14):5286.
- Hinton, G. E., ve R. R. Salakhutdinov. (2012). "A better way to pretrain deep boltzmann machines". *Advances in Neural Information Processing Systems* 25.
- Hossain, M. A., ve M. S. A. Sajib. (2019). "Classification of image using convolutional neural network (CNN)". *Global Journal of Computer Science and Technology* 19(2):13-14.
- Huang, G.-B., Q.-Y. Zhu, ve C.-K. Siew. (2004). "Extreme learning machine: a new learning scheme of feedforward neural networks". S. 985-90 içinde *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*. C. 2. Ieee.

- Huang, Y., P. Zhou, Y. Yang, T. Chen, ve N. Li. (2021). "Time-delayed reservoir computing based on a two-element phased laser array for image identification". *IEEE Photonics Journal* 13(5):1-9.
- Hussain, K., M. N. Mohd Salleh, S. Cheng, ve Y. Shi. (2019). "Metaheuristic research: a comprehensive survey". *Artificial intelligence review* 52(4):2191-2233.
- Hussain, K., Y. Xia, A. N. Onaizah, T. Manzoor, ve K. Jalil. (2022). "Hybrid of WOA-ABC and proposed CNN for intrusion detection system in wireless sensor networks". *Optik* 271:170145.
- Inik, Ö. (2025). "SwarmCNN: An efficient method for CNN hyperparameter optimization using PSO and ABC metaheuristic algorithms". *The Journal of Supercomputing* 81(8):874.
- Ioffe, S., ve C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". S. 448-56 içinde *International conference on machine learning*. pmlr.
- Isong, N. (2025). "Building Efficient Lightweight CNN Models". *arXiv preprint arXiv:2501.15547*.
- Janiesch, C., P. Zschech, ve K. Heinrich. (2021). "Machine learning and deep learning". *Electronic markets* 31(3):685-95.
- Jian, Z., W. Li, X. Tang, Y. Liang, R. Zhao, J. Tang, Y. Jiang, X. Guo, G. Tang, ve K. Yan. (2024). "Artificial photoelectric synaptic devices with ferroelectric diode effect for high-performance neuromorphic computing". *Surfaces and Interfaces* 55:105407.
- Jogin, M., M. S. Madhulika, G. D. Divya, R. K. Meghana, ve S. Apoorva. (2018). "Feature extraction using convolution neural networks (CNN) and deep learning". S. 2319-23 içinde *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*. IEEE.
- Kadam, S. S., A. C. Adamuthe, ve A. B. Patil. (2020). "CNN model for image classification on MNIST and fashion-MNIST dataset". *Journal of scientific research* 64(2):374-84.
- Karaboga, D. (2005). "An idea based on honey bee swarm for numerical optimization".
- Karaboga, D., ve B. Akay. (2011). "A modified artificial bee colony (ABC) algorithm for constrained optimization problems". *Applied soft computing* 11(3):3021-31.

- Karaboga, D., ve B. Basturk. (2007). "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm". *Journal of global optimization* 39(3):459-71.
- Karaboğa, D. (2004). *Yapay zeka optimizasyon algoritmaları*. Nobel Akademik Yayıncılık.
- Katılmış, Z., ve C. Karakuzu. (2021). "ELM based two-handed dynamic turkish sign language (TSL) word recognition". *Expert Systems with Applications* 182:115213.
- Kaur, R., R. K. Roul, ve S. Batra. (2023). "Multilayer extreme learning machine: a systematic review". *Multimedia Tools and Applications* 82(26):40269-307.
- Kaveh, M., ve M. S. Mesgari. (2023). "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review". *Neural Processing Letters* 55(4):4519-4622.
- Kervadec, H., J. Dolz, M. Tang, E. Granger, Y. Boykov, ve I. Ben Ayed. (2019). "Constrained-CNN losses for weakly supervised segmentation". *Medical image analysis* 54:88-99.
- Kervadec, H., J. Dolz, J. Yuan, C. Desrosiers, E. Granger, ve I. Ben Ayed. (2022). "Constrained deep networks: Lagrangian optimization via log-barrier extensions". S. 962-66 içinde *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE.
- Ketkar, Nikhil, Jojo Moolayil, N Ketkar, ve J Moolayil. (2021). *Deep learning with Python: learn best practices of deep learning models with PyTorch*. Springer.
- Khan, S. U. R., M. Zhao, S. Asif, ve X. Chen. (2024). "Hybrid-NET: a fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis". *International Journal of Imaging Systems and Technology* 34(1):e22975.
- Krizhevsky, A., ve G. Hinton. (2009). "Learning multiple layers of features from tiny images".
- Krizhevsky, A., I. Sutskever, ve G. E. Hinton. (2012). "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems* 25.
- Kshatri, S. S., ve D. Singh. (2023). "Convolutional neural network in medical image analysis: a review". *Archives of Computational Methods in Engineering* 30(4):2793-2810.
- LeCun, Y. (1998a). "The MNIST database of handwritten digits". [http://yann. lecun. com/exdb/mnist/](http://yann.lecun.com/exdb/mnist/).
- LeCun, Y. (1998b). "The MNIST database of handwritten digits". [http://yann. lecun. com/exdb/mnist/](http://yann.lecun.com/exdb/mnist/).

com/exdb/mnist/

- LeCun, Y., Y. Bengio, ve G. Hinton. (2015). "Deep learning". *nature* 521(7553):436-44.
- Li, J., P. Lefevre, ve A. P. P. A. Majeed. (2024). "Randomness and Interpolation Improve Gradient Descent: A Simple Exploration in CIFAR Datasets". S. 56-59 içinde *2024 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE.
- Li, J., T. Liu, X. Wang, ve J. Yu. (2022). "Automated asphalt pavement damage rate detection based on optimized GA-CNN". *Automation in Construction* 136:104180.
- Liu, G., K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, A. Tao, ve B. Catanzaro. (2018). "Partial convolution based padding". *arXiv preprint arXiv:1811.11718*.
- López-Ortiz, E. J., M. Perea-Trigo, L. M. Soria-Morillo, F. Sancho-Caparrini, ve J. J. Vegas-Olmos. (2024). "Exploring deep echo state networks for image classification: A multi-reservoir approach". *Neural Computing and Applications* 36(20):11901-18.
- Lu, S., Z. Lu, J. Yang, M. Yang, ve S. Wang. (2018). "A pathological brain detection system based on kernel based ELM". *Multimedia tools and applications* 77(3):3715-28.
- Madakam, S., T. Uchiya, S. Mark, ve Y. Lurie. (2022). "Artificial intelligence, machine learning and deep learning (literature: review and metrics)". *Asia-Pacific Journal of Management Research and Innovation* 18(1-2):7-23.
- Mavaie, P., L. Holder, ve M. K. Skinner. (2023). "Hybrid deep learning approach to improve classification of low-volume high-dimensional data". *BMC bioinformatics* 24(1):419.
- McCarthy, J., M. L. Minsky, N. Rochester, ve C. E. Shannon. (2006). "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955". *AI magazine* 27(4):12.
- Mitchell, T. M., J. G. Carbonell, ve R. S. Michalski. (1986). *Machine learning: a guide to current research*. C. 12. Springer Science & Business Media.
- Nagaveni, V. L., D. R. Rao, ve S. Neeharika. (2024). "ABC Optimised CNN model for classifying Alzheimer's Disease stages". S. 1-8 içinde *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. IEEE.

- Naidu, G., T. Zuva, ve E. M. Sibanda. (2023). "A review of evaluation metrics in machine learning algorithms". S. 15-25 içinde *Computer science on-line conference*. Springer.
- Narkhede, M. V, P. P. Bartakke, ve M. S. Sutaone. (2022). "A review on weight initialization strategies for neural networks". *Artificial intelligence review* 55(1):291-322.
- Naseri, H., ve V. Mehrdad. (2023). "Novel CNN with investigation on accuracy by modifying stride, padding, kernel size and filter numbers". *Multimedia Tools and Applications* 82(15):23673-91.
- Obaid, K. B., S. Zeebaree, ve O. M. Ahmed. (2020). "Deep learning models based on image classification: a review". *International Journal of Science and Business* 4(11):75-81.
- Obi, J. C. (2023). "A comparative study of several classification metrics and their performances on data". *World Journal of Advanced Engineering Technology and Sciences* 8(1):308-14.
- Oh, Y., H. Noh, ve W. Shin. (2021). "C-CNNLoc: Constrained CNN for robust indoor localization with building boundary". *Electronics Letters* 57(10):422-25.
- O'shea, K., ve R. Nash. (2015a). "An introduction to convolutional neural networks". *arXiv preprint arXiv:1511.08458*.
- O'shea, K., ve R. Nash. (2015b). "An introduction to convolutional neural networks". *arXiv preprint arXiv:1511.08458*.
- Özkaraca, O., O. İ. Bağrıaçık, H. Gürüler, F. Khan, J. Hussain, J. Khan, ve U. E. Laila. (2023). "Multiple brain tumor classification with dense CNN architecture using brain MRI images". *Life* 13(2):349.
- Pathak, D., P. Krahenbuhl, ve T. Darrell. (2015). "Constrained convolutional neural networks for weakly supervised segmentation". S. 1796-1804 içinde *Proceedings of the IEEE international conference on computer vision*.
- Purwono, P., A. Ma'arif, W. Rahmaniari, H. I. K. Fathurrahman, A. Z. K. Frisky, ve Q. M. ul Haq. (2022). "Understanding of convolutional neural network (cnn): A review". *International Journal of Robotics and Control Systems* 2(4):739-48.
- Rahman, T., ve M. S. Islam. (2023). "MRI brain tumor detection and classification using parallel deep convolutional neural networks". *Measurement: Sensors* 26:100694.
- Rosenblatt, F. (1958). "The perceptron: a probabilistic model for information storage and

- organization in the brain.” *Psychological review* 65(6):386.
- Rosero-Montalvo, P. D., P. Tözün, ve W. Hernandez. (2024). “Optimized CNN architectures benchmarking in hardware-constrained edge devices in IoT environments”. *IEEE Internet of Things Journal* 11(11):20357-66.
- Rumelhart, D. E., G. E. Hinton, ve R. J. Williams. (1986). “Learning representations by back-propagating errors”. *nature* 323(6088):533-36.
- Saranirad, V., S. Dora, T. M. McGinnity, ve D. Coyle. (2024). “CDNA-SNN: a new spiking neural network for pattern classification using neuronal assemblies”. *IEEE Transactions on Neural Networks and Learning Systems* 36(2):2274-87.
- Saranya, M. S., ve P. Geetha. (2024). “Prediction of Most Confident Labelled Output for Fashion Image using Deep Learning Model with Tensorflow”. S. 1-6 içinde *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE.
- Sathyanarayanan, S., ve B. R. Tantri. (2024). “Confusion matrix-based performance evaluation metrics”. *African Journal of Biomedical Research* 27(4S):4023-31.
- Shami, T. M., D. Grace, A. Burr, ve P. D. Mitchell. (2024). “Single candidate optimizer: a novel optimization algorithm”. *Evolutionary Intelligence* 17(2):863-87.
- Shamshad, N., D. Sarwr, A. Almogren, K. Saleem, A. Munawar, A. U. Rehman, ve S. Bharany. (2024). “Enhancing brain tumor classification by a comprehensive study on transfer learning techniques and model efficiency using mri datasets”. *IEEE Access*.
- Shinde, P. P., ve S. Shah. (2018). “A Review of Machine Learning and Deep Learning Applications”. S. 1-6 içinde *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*.
- Sumera, S. R., N. Anjum, ve K. Vaidehi. (2022). “Implementation of CNN and ANN for Fashion-MNIST-Dataset using Different Optimizers”. *Indian Journal of Science and Technology* 15(47):2639-45.
- Sun, M., Z. Song, X. Jiang, J. Pan, ve Y. Pang. (2017). “Learning pooling for convolutional neural network”. *Neurocomputing* 224:96-104.
- Sun, Y., G. Shi, W. Dong, ve X. Xie. (2023). “MADPL-net: Multi-layer attention dictionary pair learning network for image classification”. *Journal of Visual Communication and*

Image Representation 90:103728.

- Teow, M. Y. W. (2020). “Experimenting deep convolutional visual feature learning using compositional subspace representation and fashion-MNIST”. S. 1-6 içinde *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE.
- Tunç, A., Ş. Taşdemir, ve T. Sağ. (2022). “Comparison of heuristic and metaheuristic algorithms”. S. 76-81 içinde *2022 7th International conference on computer science and engineering (UBMK)*. IEEE.
- Turing, A. M. (2007). “Computing machinery and intelligence”. S. 23-65 içinde *Parsing the Turing test: Philosophical and methodological issues in the quest for the thinking computer*. Springer.
- Vankdothu, R., ve M. A. Hameed. (2022). “Brain tumor MRI images identification and classification based on the recurrent convolutional neural network”. *Measurement: Sensors* 24:100412.
- Venkataravanappa, V., R. A. Chowdappa, M. Shamanna, M. Krishnappa, B. Mariyappa, ve A. K. Singh. (2024a). “Conquering fashion MNIST with CNNs using computer vision by pretrained models: VGG19 and RESNET50”. S. 020010 içinde *AIP Conference Proceedings*. C. 3131. AIP Publishing LLC.
- Venkataravanappa, V., R. A. Chowdappa, M. Shamanna, M. Krishnappa, B. Mariyappa, ve A. K. Singh. (2024b). “Conquering fashion MNIST with CNNs using computer vision by pretrained models: VGG19 and RESNET50”. S. 020010 içinde *AIP Conference Proceedings*. C. 3131. AIP Publishing LLC.
- Wang, Y., H. Zhang, ve G. Zhang. (2019). “CPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks”. *Swarm and Evolutionary Computation* 49:114-23.
- Wang, Z. J., R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, ve D. H. P. Chau. (2020). “CNN explainer: learning convolutional neural networks with interactive visualization”. *IEEE Transactions on Visualization and Computer Graphics* 27(2):1396-1406.
- Weng, Q., Z. Mao, J. Lin, ve X. Liao. (2018). “Land-use scene classification based on a CNN

- using a constrained extreme learning machine”. *International journal of remote sensing* 39(19):6281-99.
- Wu, J., B. Liu, H. Zhang, S. He, ve Q. Yang. (2021). “Fault detection based on fully convolutional networks (FCN)”. *Journal of Marine Science and Engineering* 9(3):259.
- Wu, X., Z. Lin, J. Deng, J. Li, ve Y. Feng. (2024). “Nonmasking-based reservoir computing with a single dynamic memristor for image recognition”. *Nonlinear Dynamics* 112(8):6663-78.
- Wu, Y., L. Qiu, J. Wang, ve S. Feng. (2025). “The use of convolutional neural networks for abnormal behavior recognition in crowd scenes”. *Information Processing & Management* 62(1):103880.
- Xiao, H., K. Rasul, ve R. Vollgraf. (2017). “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. *arXiv preprint arXiv:1708.07747*.
- Yacouby, R., ve D. Axman. (2020). “Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models”. S. 79-91 içinde *Proceedings of the first workshop on evaluation and comparison of NLP systems*.
- Yalçın, S., S. Panchal, ve M. S. Herdem. (2022). “A CNN-ABC model for estimation and optimization of heat generation rate and voltage distributions of lithium-ion batteries for electric vehicles”. *International Journal of Heat and Mass Transfer* 199:123486.
- Yan, B., W. Zheng, D. Tang, Y. LaiLi, ve Y. Xing. (2023). “A knowledge-constrained CNN-BiLSTM model for lithium-ion batteries state-of-charge estimation”. *Microelectronics Reliability* 150:115112.
- Yang, C., H. Li, F. Lin, B. Jiang, ve H. Zhao. (2019). “Constrained R-CNN: A general image manipulation detection model”. *arXiv preprint arXiv:1911.08217*.
- Yildiz, A. R. (2012). “A comparative study of population-based optimization algorithms for turning operations”. *Information Sciences* 210:81-88.
- Yuzgec, U. (2025). “Accelerated opposition learning based chaotic single candidate optimization algorithm: A new alternative to population-based heuristics”. *Knowledge-Based Systems* 314:113169.
- Zafar, A., M. Aamir, N. Mohd Nawi, A. Arshad, S. Riaz, A. Alruban, A. K. Dutta, ve S. Almotairi. (2022a). “A comparison of pooling methods for convolutional neural

networks”. *Applied Sciences* 12(17):8643.

Zafar, A., M. Aamir, N. Mohd Nawi, A. Arshad, S. Riaz, A. Alruban, A. K. Dutta, ve S. Almotairi. (2022b). “A comparison of pooling methods for convolutional neural networks”. *Applied Sciences* 12(17):8643.

Zhang, P., Y. Zhong, ve X. Li. (2020). “ACCL: Adversarial constrained-CNN loss for weakly supervised medical image segmentation”. *arXiv preprint arXiv:2005.00328*.

Zhang, X.-C., J.-C. Yi, G.-P. Yang, C.-K. Wu, T.-J. Hou, ve D.-S. Cao. (2022). “ABC-Net: a divide-and-conquer based deep learning architecture for SMILES recognition from molecular images”. *Briefings in bioinformatics* 23(2):bbac033.

Zhu, W., J. Miao, ve L. Qing. (2015). “Constrained extreme learning machines: A study on classification cases”. *arXiv preprint arXiv:1501.06115*.