



**T.C.**

**ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**MISIR TOHUMLARININ EVRİŞİMSEL SİNİR AĞLARI VE  
HİPERPARAMETRE OPTİMİZASYONU KULLANILARAK  
HAPLOİD VE DİPLOİD OLARAK SINIFLANDIRILMASI**

**YÜKSEK LİSANS TEZİ**

**SERTUĞ FİDAN**

**Tez Danışmanı**

**Dr. Öğr. Üyesi ALİ MURAT TİRYAKİ**

**ÇANAKKALE – 2023**





T.C.

ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**MISIR TOHUMLARININ EVRİŞİMSEL SİNİR AĞLARI VE  
HİPERPARAMETRE OPTİMİZASYONU KULLANILARAK HAPLOİD VE  
DİPLOİD OLARAK SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

SERTUĞ FİDAN

Tez Danışmanı

Dr. Öğr. Üyesi ALİ MURAT TİRYAKİ

ÇANAKKALE – 2023

## ETİK BEYAN

Çanakkale Onsekiz Mart Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kuralları'na uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarımı kabullendiğimi taahhüt ve beyan ederim.

Sertuğ FİDAN

10/08/2023

## TEŐEKKÜR

Bu tez alıőması TUBİTAK 1071-Uluslararası Araőtırma Fonlarından Yararlanma Kapasitesinin ve Uluslararası Ar-Ge İőbirliklerine Katılımın Arttırılmasına Yönelik Destek Programı ile desteklenen 221N269 nolu (Alt Proje No: TUBİTAK 221N418) proje kapsamında gerekleőtirilmiőtir. Desteklerinden dolayı TÜBİTAK'a teőekkür ederim. Tezin sonulandırılması aőamasında bilgi ve tecrübesiyle alıőmam boyunca her an bir elini omzumda olduėunu hissettiren kıymetli danıőman hocam Dr. Ali Murat TİRYAKİ'ye, eėitim hayatım boyunca bana güvenlerini ve inanlarını kaybetmeden destek gösteren ve beni bu günlere getiren babam Enver FİDAN'a ve annem Hülya FİDAN'a sonsuz teőekkürlerimi sunarım.

Sertuė FİDAN  
anakkale, Aėustos 2023

## ÖZET

# Mısır Tohumlarının Evrişimsel Sinir Ağları ve Hiperparametre Optimizasyonu Kullanılarak Haploid ve Diploid Olarak Sınıflandırılması

Sertuğ FİDAN

Çanakkale Onsekiz Mart Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı Tezi

Danışman: Dr. Öğr. Üyesi Ali Murat TİRYAKI

31/08/2023, 68

Üretimi kolay olması, özellikle hayvan yemi olarak kullanılabilme olanağı yüzünden mısırın verimli bir şekilde üretimi toplumumuzun sürdürülebilirliği açısından büyük önem taşımaktadır. Günümüzde mısır ıslahı için kullanılan en popüler yöntem Katlanmış Haploid (Doubled Haploid, DH)'tir. Bu yöntemin kullanılmasında en önemli adımlardan birisi haploid tohumların, diploid tohumlardan ayrılmasıdır. Bu sınıflandırma işlemi genellikle eğitilmiş teknisyenler tarafından yapılmaktadır. Makine öğrenmesi yöntemlerindeki gelişmeler sonucunda bu işlem çeşitli sensör verilerinden yararlanılarak otomize edilebilir hale gelmiştir. Bu çalışmada, sınıflandırma işlemi gerçekleştirilmek için 1230 haploid ve 1770 diploid mısır tohumu görüntüsünden yararlanılarak yeni bir Evrişimli Sinir Ağı (Convolutional Neural Network, CNN) model mimarisi geliştirilmiştir. Bu modelin hiperparametreleri Ağaç Bazlı Parzen Tahminleyicileri (Tree Based Parzen Estimators, TPE) algoritmasından yararlanarak optimize edilmiştir. Üretilen yeni model eğitilebilen parametre sayısı ve doğrulama skoru bakımından (9 milyon parametre %94,66 doğruluk) VGG-19 modeline kıyasla (144 milyon parametre, %94,22 doğruluk) tohum sınıflamasında daha başarılı sonuç elde etmiştir.

**Anahtar Kelimeler:** Hiperparametre optimizasyonu, Evrişimli Sinir Ağları, Ağaç Yapılı Parzen Tahminleyicileri, Katlanmış Haploid.

## ABSTRACT

### **Classification of Corn Seeds as Haploid and Diploid Using Convolutional Neural Networks and Hyperparameter Optimization**

Sertuğ FİDAN

Çanakkale Onsekiz Mart University

School of Graduate Studies

Master of Science Thesis in Computer Engineering

Advisor : Dr. Öğr. Üyesi Ali Murat TİRYAKİ

31/08/2023, 68

The efficient production of corn is of great importance for the sustainability of our society, due to its ease of production and the possibility of using it as animal feed. The most popular method used for maize breeding today is Doubled Haploid (DH). One of the most important steps in using this method is the separation of haploid seeds and diploid seeds. This classification process is usually performed by trained technicians. As a result of the developments in machine learning methods, this process has become automated by using various sensor data. In this study, a new Convolutional Neural Network (CNN) model architecture was developed using 1230 haploid and 1770 diploid corn seed images to perform the classification process. The hyperparameters of this model have been optimized using the Tree Based Parzen Estimators (TPE) algorithm. The new model produced was more successful in seed classification compared to the VGG-19 model (144 million parameters, 94.22% accuracy) in terms of the number of trainable parameters and validation score (9 million parameters 94.66% accuracy).

**Keywords:** Hyperparameter optimization, Convolutional Neural Networks, Tree Based Parzen Estimators, Doubled Haploid.

# İÇİNDEKİLER

## Sayfa No

JÜRİ ONAY SAYFASI.....	i
ETİK BEYAN.....	ii
TEŞEKKÜR.....	iii
ÖZET .....	iv
ABSTRACT .....	v
İÇİNDEKİLER .....	vi
SİMGELER ve KISALTMALAR.....	viii
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ.....	x

## BİRİNCİ BÖLÜM

### GİRİŞ

1

## İKİNCİ BÖLÜM

### ÖNCEKİ ÇALIŞMALAR

5

2.1. Makine öğrenmesi.....	5
2.1.1. Derin öğrenme .....	6
2.2. Evrişimli Sinir Ağları .....	9
2.2.1. Evrişimli Sinir Ağı Katmanları.....	11
Evrişim Katmanı.....	11
Havuzlama Katmanı.....	13
Tam Bağlı Katman.....	13
Sönümleme Katmanı.....	14
Toplu Normalleştirme .....	14
2.3. Optimizasyon Algoritmaları .....	15
2.3.1. Karelerin Ortalamasının Karekök Yayılımı .....	15
2.3.2. Adaptif Moment Tahmini .....	16
2.3.3. Düzeltilmiş Adaptif Moment Tahmini .....	18
2.4.4. Nesterov ile Hızlandırılmış Uyarlanabilir Moment Tahmini .....	18



2.4. Mısır Tohum Sınıflandırması .....	19
2.5. Hiperparametre Optimizasyonu .....	25
<b>ÜÇÜNCÜ BÖLÜM</b>	
<b>MATERYAL YÖNTEM</b>	
3.1. Çalışmada Kullanılan Yazılım ve Donanım Altyapısı .....	27
3.2. Veri kümesi .....	27
3.3. Hyperopt Stokastik Parametre İfadeleri .....	28
3.4. Hiperparametre Uzayı .....	31
3.5. Ağaç Bazlı Parzen Tahminleyicileri .....	36
3.6. Başlangıç Parametreleri .....	37
3.7. Ön İşleme Aşaması ve Eğitim Prosedürü .....	38
3.8. Modellerin Üretim Adımları .....	39
<b>DÖRDÜNCÜ BÖLÜM</b>	
<b>ARAŞTIRMA BULGULARI</b>	
<b>BEŞİNCİ BÖLÜM</b>	
<b>SONUÇ ve ÖNERİLER</b>	
KAYNAKÇA .....	62
ÖZGEÇMİŞ .....	I

## SİMGELER VE KISALTMALAR

ANN	Artificial Neural Networks (Yapay Sinir Ağları)
CNN	Convolutional Neural Network (Evrışimli Sinir Ağları)
ReLU	Rectified Linear Unit (Doğrultulmuş Doğrusal Birim)
VGG-19	Visual Geometry Group-19 (Görsel Geometri Grubu-19)
MAS	Marker Assisted Selection (İşaretleyici Yardımlı Seçim)
ANN	Artificial Neural Networks (Yapay Sinir Ağları)
TPE	Tree-Structured Parzen Estimators (Ağaç Bazlı Parzen Tahminleyicileri)
DL	Deep Learning (Derin Öğrenme)
HPC	High Performance Computing (Yüksek Performans Bilgi İşlem)
AÖ	Discriminative Learning (Ayrımcı Öğrenme)
GL	Generative Learning (Üretici Öğrenme)
ReLU	Rectified Linear Unit (Doğrultulmuş Lineer Birim)
BN	Batch Normalization (Toplu Normalleştirme)
SVM	Support Vector Machines (Destek Vektörü Makineleri)
RF	Random Forest (Rastgele Ağaç)
LR	Logistic Regression (Lojistik Regresyon)
NIR	Near Infrared (Yakın kızılötesi)
SVSKLPP	Supervised Virtual Sample Kernel Locality Preserving Projection (Denetimli Sanal Örnek Çekirdek Yeri Koruma Projeksiyonu)
ZCA	Zero-phase Component Analysis (Sıfır Düzeyli Bileşen Analizi)
PLSE	Partial Least Squares Estimation (Kısmi En Küçük Kareler Regresyonu)
BPNN	Back Propagation Neural Network (Geri Yayılımlı Sinir Ağı)
BULDP	Biomimetic Uncorrelated Locality Discriminant Projection (Biyomimetik İlişkisiz Yerel Ayrımcı Projeksiyonu)
COR	Correct Acceptance Rate (Doğru Kabul Oranı)
PNN	Probabilistic Neural Network (Olasılıksal Sinir Ağı)
DT	Decision Trees (Karar Ağaçları)
ESA	Evrışimli Sinir Ağları
MLP	Çok Katmanlı Algılayıcı (Multilayer Perceptron)

## TABLULAR DİZİNİ

<b>Tablo No</b>	<b>Tablo Adı</b>	<b>Sayfa No</b>
<b>Tablo 1</b>	Modellerin üretimi aşamasında kullanılan başlangıç parametreleri ve bu parametreleri etkileyen hiperparametreler tablosu	38
<b>Tablo 2</b>	En iyi modele ait hiperparametre değerleri tablosu	56



## ŞEKİLLER DİZİNİ

Şekil No	Şekil Adı	Sayfa No
Şekil 1	Derin öğrenmenin konumunun örneği (Sarker,2021)	7
Şekil 2	Büyük miktarda veri için derin öğrenme ve makine öğrenmesi arası performans karşılaştırması örneği (Sarker,2021)	8
Şekil 3	Derin öğrenme yöntemleri kategorileri (Sarker,2021)	9
Şekil 4	Beş katmanlı ESA mimarisi örneği (O'Shea ve Nash, 2015)	11
Şekil 5	Evrşim katmanında her adımda gerçekleşen hesaplamalar (Alzubaidi vd., 2021)	12
Şekil 6	(a) Diploid mısır tohumu örneği, (b) Haploid mısır tohumu örneği (Altuntaş, Cömert vd., 2019)	28
Şekil 7	Modellerin üretim adımları	39
Şekil 8	Üretilen tüm modeller için adım sayısına karşılık gelen test doğruluğu için öğrenme eğrileri	41
Şekil 9	En iyi 20 model için batch_size hiperparametresi sonuçları	42
Şekil 10	En iyi 20 model için activation hiperparametresi sonuçları	43
Şekil 11	En iyi 20 model için conv_kernel_size hiperparametresi sonuçları	44
Şekil 12	En iyi 20 model için conv_pool_res_start_idx hiperparametresi sonuçları	44
Şekil 13	En iyi 20 model için conv_dropout_proba hiperparametresi sonuçları	45
Şekil 14	En iyi 20 model için conv_hiddn_units_mult hiperparametresi sonuçları	46
Şekil 15	En iyi 20 model için fc_dropout_drop_proba hiperparametresi sonuçları	47
Şekil 16	En iyi 20 model için fc_units_l_mult hiperparametresi sonuçları	48
Şekil 17	En iyi 20 model için first_conv hiperparametresi sonuçları	48
Şekil 18	En iyi 20 model için l2_weight_reg_mult hiperparametresi sonuçları	49

<b>Şekil 19</b>	En iyi 20 model için lr_rate_mult hiperparametresi sonuçları	50
<b>Şekil 20</b>	En iyi 20 model için nb_conv_pool_layers hiperparametresi sonuçları	50
<b>Şekil 21</b>	En iyi 20 model için one_more_fc hiperparametresi sonuçları	51
<b>Şekil 22</b>	En iyi 20 model için optimizer hiperparametresi sonuçları	52
<b>Şekil 23</b>	En iyi 20 model için pooling_type hiperparametresi sonuçları	52
<b>Şekil 24</b>	En iyi 20 model için res_conv_kernel_size hiperparametresi sonuçları	53
<b>Şekil 25</b>	En iyi 20 model için residual hiperparametresi sonuçları	54
<b>Şekil 26</b>	En iyi 20 model için use_BN hiperparametresi sonuçları	54
<b>Şekil 27</b>	Tüm modeller için hiperparametre uzayı üzerinde denenmiş parametrelerin test doğruluğuna bağlı dağılımını gösteren matris grafiği	55
<b>Şekil 28</b>	Üretilen en iyi modele ait mimarinin yapısı	57

## BİRİNCİ BÖLÜM

### GİRİŞ

Dünyada buğdaydan sonra en fazla üretilen tahıl türü mısırdır. Mısır üretimi için kullanılan alanın %64'ü gelişmekte olan ülkelerde bulunmaktadır. Buna rağmen toplam ürün hasadının sadece %43'ü bu bölgelerde gerçekleştirilmektedir. Bu durumun genel sebebi gelişmiş ülkelerde hektar başına düşen ortalama verimin, gelişmekte olan ülkelere göre daha yüksek olmasıdır. Gelişmiş ülkeler, gelişmekte olan ülkelere oranla aynı miktar alanda 2 kattan daha fazla verimlilik ile mısır üretimi gerçekleştirebilmektedir. Bu farkın sebepleri çevresel, teknolojik ve organizasyonel faktörlerden dolayı oluşmaktadır (Dowswell, 2019). Bu araştırma ile ilgilenilen konu teknolojik faktörlerden biri olan hibrit mısırların kullanımınıdır.

Küresel iklim değişikliğinin yaratacağı en önemli problemlerden biri yağış rejimlerinde oluşması beklenen düzensizliklerdir. Bu düzensizliğin bazı tarımsal ürünlerin belirli bölge ve arazilerde üretim devamlılığını kaybetmesine veya rekoltede büyük düşüşler görülmesine yol açması beklenmektedir. Genel anlamda sıcaklığın artması ile birlikte havanın su tutma kapasitesi her bir derecelik artış için %7 oranında artacağı için atmosferdeki su buharının önümüzdeki senelerde gittikçe yükselmesi beklenmektedir (WorldBank, 2022). Bu kırılganlığın sebepleri olarak ulaşım sisteminin yeterince iklimden bağımsız olmaması, gıda güvenliği sorunları, artan su stresi ve 2021 yılında gerçekleşen orman yangınları gösterilmektedir. Atmosferde ki su buharının artması ile ülkemizde fırtınaların çok daha etkili olabileceği öngörülmektedir. Sıcaklığın yükselmesi ile bölgelere yağması beklenen kar miktarının azalması, yağmur yağışlarının ise artması beklenmektedir. Genel kanı, kurak olmaya yakın bölgelerin daha da kuraklaşacağı düzenli yağış alan bölgelerin ise çok daha fazla yağış alacağı yönündedir. Dolayısıyla ülkemiz için kuraklığa ve aşırı yağışa dayanıklı tohum üretilmesi büyük önem taşımaktadır.

Bitki ıslah programlarının genel amacı, tohumlarda oluşacak uygun alel (gen) kombinasyonlarının sıklığını arttırmaktır. Mısır ıslahı için tarih boyunca uygulanan genel yöntem, popülasyondaki üstün bireylerin seçimi, bu bireylerin kendi aralarında melezlenmesi ve sonrasında oluşan popülasyon içerisinde yine üstün bireylerin seçilerek önceki adımların tekrar edilmesi işlemidir. Bu yöntem, tohum çeşitlerinin ortama adapte

olmasını büyük ölçüde yardımcı olsa da verimliliği yüksek çeşitlerin üretilmesine uygun değildir. Yirminci yüzyılda kendilenmiş-hibrit kavramı gelişene kadar bu yöntem kullanılmaya devam edilmiştir (Hallauer ve M.J. Carena, 2009). Kendilenmiş-hibrit yöntemi ile istenilen ortama uygun ve hasat verimliliği daha yüksek tohumlar üretilebilir hale gelmiştir.

Kendilenmiş-hibrit yöntemi ile istenilen homozigotluk düzeyine ulaşmak için altı ile sekiz nesillik kendileme ve seçilim işlemine gerek duyulmaktadır (M. Carena vd., 2010). Üretilen çeşitlerin tescili de göz önünde bulundurulursa bu işlem yaklaşık olarak 11-13 yıl süre alması beklenmektedir (Chaikam vd., 2019). Günümüzde bu sürenin düşürülebilmesi için alternatif yöntem olarak Katlanmış Haploid (Doubled Haploid, DH) yöntemi kullanılmaktadır. Bu yöntem sayesinde ıslah süresi geleneksel yöntemlere göre 6 sene kadar kısaltılabilmekte ve ıslah için harcanan zaman, maliyetten kazanç sağlanmaktadır (Bayhan vd., 2021).

DH teknolojisi sayesinde hibritlenme sürecinin hızlanmasının yanı sıra daha verimli ve doğru seçilim yapılmasına olanak sağlamakta ve tohum nakliyesi, envanter yönetimi, fidanlık dikimi ve üretim hatlarının bakımı işlemleride kolaylaşmaktadır (Prasanna vd., 2012).

Haploid hücreler, tam kromozom setine ( $2n$  kromozom, Diploid) sahip hücrenin mayoz bölünme sonucu bu setin sadece yarısına sahip ( $n$  kromozom, Haploid) olması durumunda oluşmaktadır. DH üretebilmek için, haploid hücrelerin sahip oldukları genomun kopyalanması işlemi gerçekleştirilmektedir. İkinci kromozom seti orijinali ile aynı olduğundan %100 homozigotluk (kromozomların üzerinde alleler aynı) sağlanmaktadır (homozygote, 2012). Bu homozigotluk yüzünden çaprazlandıkları zaman aynı özellikleri gösteren nesiller üretilebilmektedir. Kromozomların ikiye katlanmasını teşvik etmek için en verimli olan ve yaygın biçimde kullanılan işlem kolşisin uygulamasıdır. Bu uygulamada kullanılan kolşisin kimyasalı sayesinde haploidlerin yapay olarak kromozomları ikiye katlanmaktadır. Böylece gereken kromozom setine sahip olmadıkları için normalde üreme özelliği bulunmayan haploid hücreler, diploid hale getirilerek üreme özelliği kazandırılmaktadır (Chaikam vd., 2020).

Haploid hücreler, ya erkek yada dişi bitki genomunun sadece birine sahiptirler. Bu yüzden pratikte sadece hücrelerin belirli bir kısmı DH oluşturma kabiliyetine sahiptir. Bu kabiliyeti arttırmak için hücre içinde (in vivo) ve hücre dışında (in vitro) gerçekleştirilen yöntemler geliştirilmiştir. Hücre içinde (in vivo) bazı yöntemlerden önemli örneklerinden biri haploid indükleyicilerin kullanımınıdır (Segui-Samarro vd., 2021). Haploid indükleyiciler sayesinde DH yönteminde kullanmak için gerekli olan doğada çok az sayıda bulunan haploid tohumların sayısının artırılması hedeflenmektedir. Doğada kendiliğinden bulunma oranı %0.1 (Chaikam vd., 2019) olan haploidler, kullanılan indükleyici tipine göre %8 - %10 aralığına çıkartılabilmektedir (Prigge vd., 2012).

DH hatlarında işaret destekli seçim kullanılması (Marker Assisted Selection, MAS), allellerin daha hızlı ve verimli sabitlenmesini sağlamaktadır (Chaikam vd., 2019). MAS sayesinde hastalıklara, kuraklığa direnci yüksek, verimi yüksek tohumların oluşturulması için uygun DNA dizilerinin seçilmesi sağlanmaktadır. MAS, sayesinde geliştiriciler istenilen en uygun gen kombinasyonları seçebilmekte ve fenotipik değerlendirmeye olan ihtiyacı büyük ölçüde azaltmaktadır. Bu sayede tohum üretimi sırasında harcanan zaman, maliyet düşürülmekte ve çevresel faktörlerden oluşan istenmeyen etkiler azaltılmaktadır (Boopathi, 2020). MAS yöntemlerinden biri olan R1-Navajo (R1-nj) renk işaretleyicisi, mısırın tohum aşamasında haploid olup olmadığının hızlı ve kolay bir şekilde belirlenmesine olanak sağlamaktadır. R1-nj geni, mısır taneleri üzerinde antosiyonin pigmentasyonu üretmektedir. Bu pigmentasyon haploid mısır tohumlarının belirlenmesi için yaygın olarak kullanılmaktadır. Navajo fenotipi bazı tohum çeşitlerinde tamamen baskılanabilmekte yada zayıf bir ifade gücüne sahip olabilmektedir (Chaikam vd., 2015). R1-nj işaretleyicisine sahip mısır tohumu eğer diploid ise hem endosperm hem embriyo bölgesinde mor renklenme görülmektedir. Ancak mısır tohum haploid ise renklenme sadece endosperm bölgesinde gerçekleşmektedir (Veeramani vd., 2018). Bu bilgiden yararlanılarak eğer mısır tohumunun çekirdeğinde mor renklenme varsa tohum diploid, yoksa haploid olduğu anlaşılmaktadır.

Tohumlara ait renklenme bilgisinden yararlanarak tohumları haploid ve diploid gruplandırmak için günümüzde kullanılan en yaygın yöntem manuel incelemedir (Veeramani vd., 2018). Bu yöntemde uzmanlar her tohum için (R1-nj'den kaynaklanan) çekirdek bölgesinde renklenme olup olmadığını inceleyerek sınıflandırma işlemini



gerçekleştirir. Renklenme tohumun genetik özelliğine göre çekirdek üzerinde çok küçük bir noktada yada tohumun tamamında gerçekleşebilir. Aynı zamanda oluşan renk ifadesi yine tohumun genetik özelliklerine bağlı olarak çok açık yada koyu olabilmektedir. Bu çeşitlilikten dolayı tohumların sınıflandırılması işlemi uzmanlar tarafından gerçekleştirilirken hata payı yüksek olabilmektedir. Sınıflandırma işlemi mısır tanesi sayısı göz önünde bulundurulduğunda gerekli iş gücü açısından maliyetlidir (Melchinger, Shipprack vd., 2013). Bu sebeplerden dolayı tohumların sınıflandırılması işleminin otomatikleştirilmesi büyük önem taşımaktadır.

Literatürde bu problemin çözümü üzerine odaklanan çalışmalar incelendiğinde mısır tohumlarının sınıflandırılması için farklı yöntemlerin kullanıldığı görülmektedir. Bu yöntemler; farklı veri kümelerinden yararlanarak tohuma ait hücrenin fiziksel veya kimsiyal özelliklerinin incelenmesi (Flow sitometri, Akış Sitometrisi) yöntemi, tohuma ait floresan spektroskopisinden yararlanılması, tohumun yakın kızılötesi spektrumuna bakılması, terahertz zaman alanı spektroskopisinin incelenmesi, RGB görüntü çekilmesi sonucu tohuma ait veri elde edildikten sonra çeşitli makine öğrenmesi yöntemleri ile analiz edilmesi olarak gruplanabilir.

Bu tez çalışmasının hazırlanma amacı RGB bazlı mısır tohumu görüntüleri içeren erişime açık veri kümesinden yararlanılarak, hyperopt kütüphanesi yardımıyla doğrulama skoru açısından verimli sonuç veren Evrişimli Sinir Ağları (ESA) bazlı yeni bir derin öğrenme model mimarisi oluşturmaktır. Uygun modelin oluşturulabilmesi için hiperparametre uzayı tanımlanmış, Ağaç bazlı parzen tahminleyicileri yardımıyla parametre uzayından model mimarisini en iyileyen değerler seçilmiş bu değerlerden yararlanılarak 150 farklı model üretilmiştir. Tez çalışmasında test verisi üzerinde en iyi doğrulama skorunu elde eden 20 model arasındaki benzerlikler ve farklılıklar incelenmiştir. Çalışmada oluşturulan hiperparametre uzayının optimize edilebilmesi için gelecekte yapılabilecek güncellemelere de yer verilmektedir. Tez kapsamında literatürdeki diğer çalışmalardan farklı olarak hiperparametre optimizasyonu gerçekleştirilmiştir.

## İKİNCİ BÖLÜM

### ÖNCEKİ ÇALIŞMALAR

Bu bölümde tez çalışmasında kullanılan makine öğrenmesi, evrimsel sinir ağları, mısır tohumu sınıflandırma gibi yöntem ve teknolojiler hakkında bilgi sunulmaktadır. Günümüze kadar literatürde tanıtılmış mısır tohumlarının makine öğrenmesi tabanlı sınıflandırılması yöntemleri incelenerek bu yöntemlerin güçlü ve zayıf yönleri bölüm içerisinde tartışılmaktadır. Ayrıca, literatürde Evrimsel Sinir Ağlarında parametrelerin optimizasyonunun gerçekleşmesi için kullanılan çeşitli yaklaşımlar hakkında bilgi verilmektedir.

#### 2.1. Makine Öğrenmesi

Makine öğrenmesi, belirli bir sonucu üretmek için programlamaya gerek duymadan girdi verilerinden yararlanılarak istenilen çıktıların üretilmesini sağlayan ve hesaplamalar içeren bir süreçtir. Makine öğrenmesi algoritması, gerçekleştirilmek istenen görevi daha iyi bir biçimde tamamlayabilmek için kendine ait mimariyi tekrarlama veya eğitim yoluyla uyarlamaktadır. Uyarlanmanın sağlanabilmesi için girdi ve bu girdilere karşılık gelen çıktılardan yararlanılarak eğitim gerçekleştirilmektedir. Algoritma eğitim sonrasında daha önce görülmemiş verilerden istenilen sonucu üretmek için kendisini yeniden yapılandırmaktadır. Bu eğitim, makine öğrenmesindeki "öğrenme" kısmına karşılık gelmektedir. Bir bilgisayar programının verilen görev sınıflarına ait performansının tecrübeyle gelişmesi durumunda o programın makine öğrenmesinden yararlandığı söylenebilmektedir (Mitchell, 1997).

Eğitim sonucunda, algoritmanın kendini uyarlamasının birçok yolu vardır. Girdi verileri seçilebilir ve ağırlıklandırılabilir, algoritma ayarlanabilme özelliğine sahip sayısal parametrelere sahip olabilir. Bu sayısal parametreler yinelemeli (iterative) optimizasyon yoluyla optimum sonuçlar üretecek şekilde düzenlenebilir. Girdi verilerinden olasılık dağılımları belirlenebilir ve sonuçları tahmin etmek için bu dağılım kullanılabilir.

Makine öğrenmesi, bir hedefe ulaşmak için insanların duyuşal sinyalleri nasıl işledikleri ve insanların bu sinyallerin işlenmesini nasıl öğrendiklerini taklit etmeyi amaçlamaktadır. Üç tür makine öğrenimi vardır (El Naqa ve Murphy, 2015):

1) Denetimli öğrenme, her öğrenme örneğinde bir girdi (renk, koku v.b.), bu girdiye ait bir etiket (elma, portakal) ile ilişkilendirilmektedir. Böylece etiket ile ilgili özelliklerin tek tek programlanması yerine etikete ait özellikler, gerçek veriler üzerinden tekrar tekrar deneyimlenerek etikete ait benzerlik ve farklılıklar sayesinde etiket ayırt edilebilir hale getirilir. Başarılı öğrenme sonucu daha önce hiç görülmemiş girdilerin uygun şekilde etiketlenebilmesi beklenir.

2) Denetimsiz öğrenme, bu yöntemde örnek olarak hedef tahtasına dart atmaya çalışan bir cihazın, çalışma prensiplerini sabit bir şekilde programlamak yerine cihazın dart atımı sonrası hedef noktasına, her deneme sonrası ne kadar yaklaştığı ölçülmesi düşünülebilir. Ölçülen değerin minimize edilmesi için cihazın parametreleri yeniden oluşturulur. Algoritma, eğitim için ayrılmış girdi verilerinden yararlanarak kendi yolunu bulur. Denetimli öğrenmeye göre en büyük farkı etiketlerin tanımlı olmamasıdır.

3) Yarı denetimli öğrenme, bu yöntemde verinin bir kısmı için etiket var, bir kısmı için yoktur. Etiketli kısım, etiketlenmemiş kısımda bulunan bilginin öğrenilmesine yardımcı olarak kullanılabilir. Bu yöntem doğadaki çoğu sürece uygundur ve insanların becerilerini geliştirmek için kullandığı süreci yakından takip etmektedir.

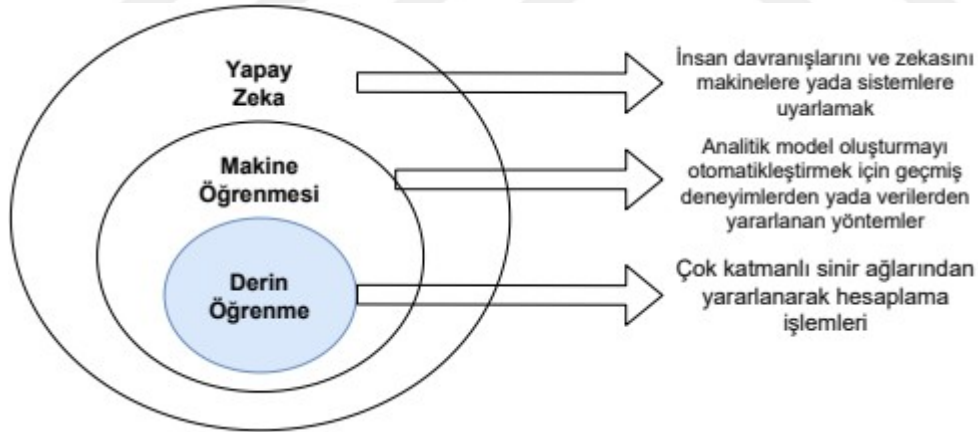
Günümüzde makine öğrenmesinden, örüntü tanıma (Bishop ve Nasrabadi, 2006), medikal uygulamalar (Cleophas vd., 2013), finans (Gyorfi vd., 2012), ekoloji (Fielding, 1999) gibi farklı alanlarda yararlanılmaktadır. Tez kapsamında denetimli öğrenme kullanılmıştır.

### **2.1.1. Derin Öğrenme**

Derin öğrenme (Deep Learning, DL), 2006 yılında (Hinton vd., 2006) öne sürülmüştür. Yapılan bu çalışma yapay sinir ağları bazlı olup rakamlara ait el yazısı görüntülerinden yararlanarak etiket tahmini gerçekleştirmektedir. Yöntem, zamanın en iyi sınıflandırma algoritmalarına oranla daha iyi rakam sınıflandırma sonuçları elde

etmiştir. Çalışma yapay sinir ağları bazlı araştırmalara olan ilgiyi arttırmış, öne çıkan bir konu haline gelmiştir (Sarker, 2021).

Şekil 1’de görülebileceği üzere derin öğrenme, yapay zeka ve makine öğrenmesi ile ilgili yöntemlerin bir alt sınıfı olarak düşünülebilir. Derin öğrenme yöntemlerinin, kendilerine ait temel özellikleri ve bağımlılıkları bulunmaktadır. Bunlardan ilki büyük veriye olan bağımlılıktır. Daha önce yapılan çalışmalar (LeCun vd., 2015) derin öğrenme yöntemlerinin, veri hacmi küçük olduğunda genellikle düşük performans ile çalıştıklarını göstermiştir. Yöntemlerin başka bir bağımlılığı ise hacimsel anlamda büyük miktarda veri ile çalışırken ihtiyaç duydukları donanım kaynaklarının maliyetidir. Geleneksel makine öğrenme yöntemlerine oranla, hesaplama işlemlerinin büyüklüğünden ötürü verimli çalışma gerçekleştirebilmek için GPU donanımına ihtiyaç duyulmaktadır. Genel anlamda derin öğrenmede, geleneksel makine öğrenme yöntemlerine oranla daha yüksek performanslı bilgi işleme (High Performance Computing, HPC) ihtiyaç olduğu söylenebilir (Coelho vd., 2017).



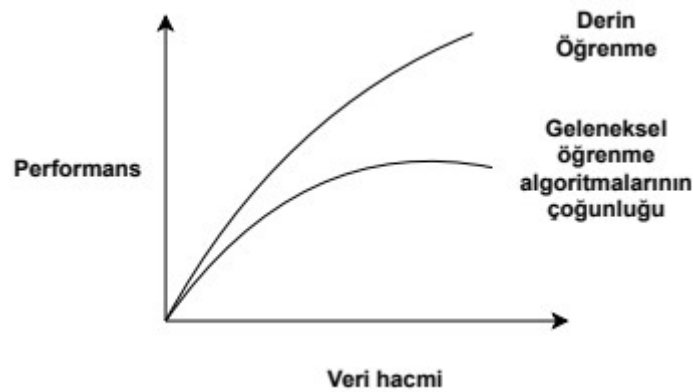
Şekil 1. Derin öğrenmenin konumunun örneği (Sarker,2021).

Derin öğrenmenin, geleneksel yöntemlere oranla en büyük avantajlarından biri derin öğrenmenin, yüksek düzeyli özneliklerin veri üzerinden otomatik olarak çıkartılabilesine olanak sağlamasıdır. Bu sayede farklı problemler için öznelik çıkarıcı oluşturmak için gereken zaman, çaba ve maliyet indirgenmiş olmaktadır (Deng, D. Yu vd., 2014). Derin öğrenme yöntemlerinin iki farklı dezavantajı bulunmaktadır. Bunlardan ilki derin öğrenme bazlı algoritmalarda kullanılan yüksek eğitilebilir parametre sayısından

ötürü eğitim işlemi geleneksel yöntemlere oranla daha uzun sürmektedir (Xin vd., 2018). İkincisi ise derin öğrenme bazlı yöntemlerin açıklanabilirliğinin düşük olmasıdır. DL, Kapalı kutu (Black Box) yapısından ötürü sonuca nasıl ulaşıldığına dair kural yapısı ve bilgi bütününe elde edilmesine olanak sağlamamaktadır (Sarker, 2021). DL'nin, geleneksel yöntemlere göre en büyük avantajı eğitim için kullanılacak verinin büyümesi ile performansın bir noktadan sonra sabit kalması yerine sürekli artmasıdır. Bu performans karşılaştırması Şekil 2 ile gösterilmiştir.

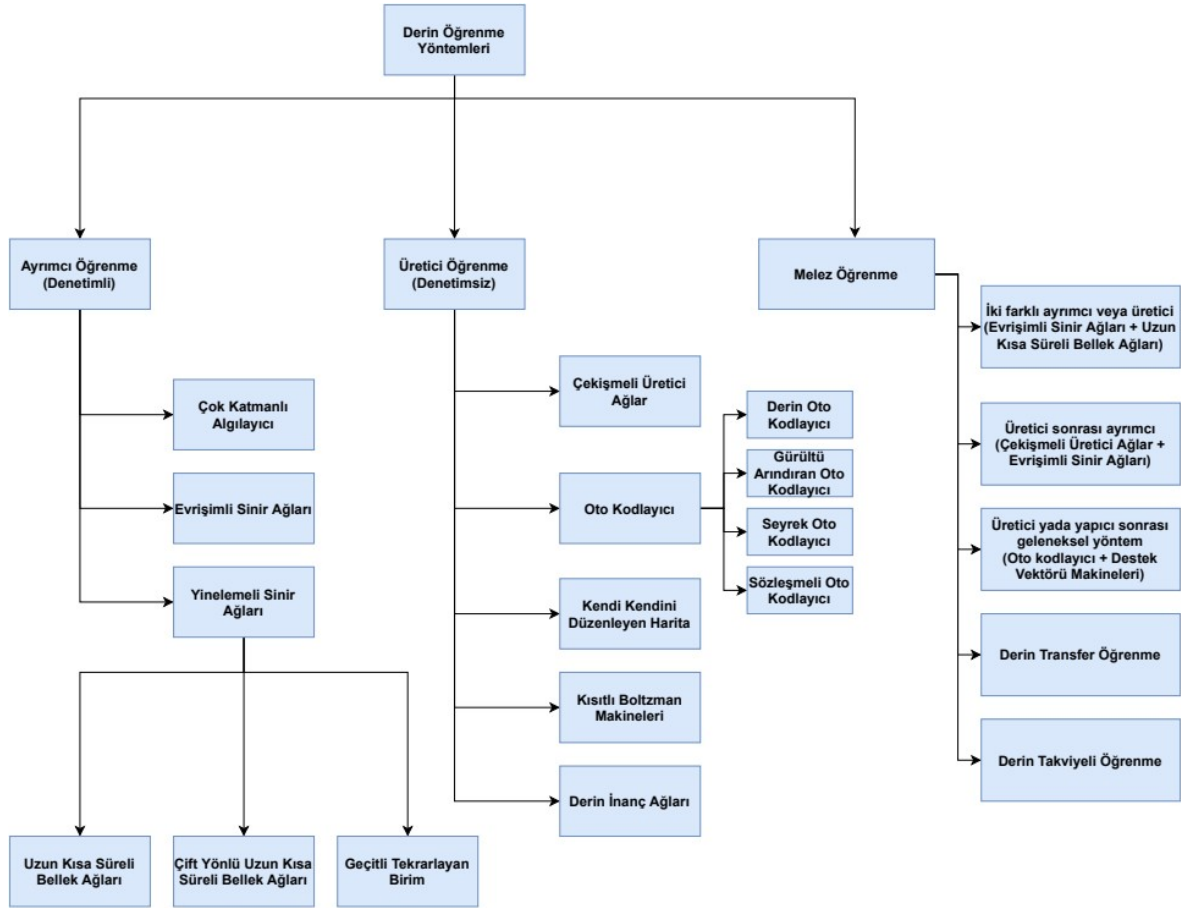
Literatürde derin öğrenmenin aşağı da durumlarda kullanılması önerilmiştir (Alzubaidi vd., 2021).

- Öğrenilecek konu hakkında insan uzmanların bulunmaması durumunda
- İnsanlar uzmanların aldıkları kararları açıklayamadığı (dili anlama, tıbbi kararlar v.b.) durumlarda
- Çözülmesi gereken probleme ait verilerin zamana bağlı olarak güncellendiği durumlarda (fiyat tahmini, stok tercihi, hava durumu tahmini)
- Problem çözümünün uyarılama (kişiselleştirme, biyometri) gerektirdiği durumlarda
- Problem boyutunun çok büyük olması ve insan muhakeme yeteneğini aşması (duygu analizi, web sayfası sıralaması hesaplaması) durumunda



Şekil 2. Büyük miktarda veri için derin öğrenme ve makine öğrenmesi arası performans karşılaştırması örneği (Sarker,2021).

Derin öğrenme yöntemleri genel anlamda üç farklı başlık altında incelenebilmektedir. Bu başlıklar: denetimli veya ayrımcı öğrenme (Discriminative Learning, AÖ), denetimsiz veya üretici öğrenme (Generative Learning, GL), melez öğrenme olarak tanımlanabilir. Kategorilerin hiyerarşisi Şekil 3 ile gösterilmiştir.



Şekil 3. Derin öğrenme yöntemleri kategorileri (Sarker,2021).

Tez ile amacımız denetimli öğrenme yöntemlerinden biri olan ESA'nın belirli bir veri kümesi için validasyon skoru yönünden en iyileştirilmesi için uygun parametrelerin bulunmasını sağlamaktır.

## 2.2. Evrişimli Sinir Ağları - ESA

ESA, öğrenme yoluyla kendine kendini optimize eden nöronlardan oluşmasından dolayı aynı yöntemi kullanan yapay sinir ağlarına benzemektedir. Her iki ağ çeşidinde de

her nöron bir girdi alacak ve bir işlem gerçekleştirecektir. Ham görüntü vektörlerinden oluşan girdiden, sınıf puanını oluşturan son çıktıya kadar ağın tamamı tek bir algısal puan işlemi (ağırlık) kullanılarak ifade edilmektedir. Son katman, her iki ağ çeşidinde de sınıflara ait kayıp fonksiyonlarını içermektedir. ESA ile yapay sinir ağları arasındaki en büyük fark ESA'nın özellikle görüntülere ait örüntülerin tanınması alanında kullanılmasıdır. Bu sayede ESA'da, görüntüye ait özellikler mimarinin içerisine kodlanılabilmekte, ağ görüntü odaklı problemlerin çözümüne daha uygun hale getirilebilmekte ve modelin oluşturulması için gerekli parametrelerin sayısı azaltılabilmektedir (O'Shea ve Nash, 2015).

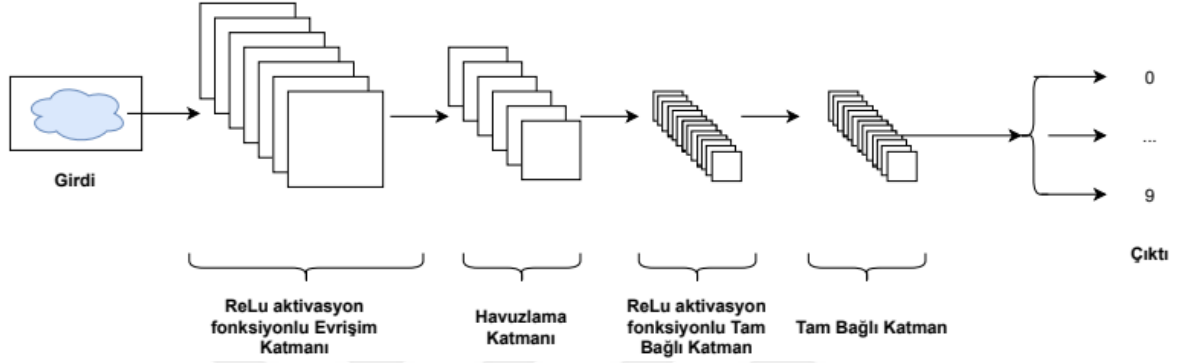
ESA'nın bazı avantajları şunlardır (Alzubaidi vd., 2021) :

- Kullanılan evrişim katmanı sayesinde görüntülere, problemin çözümüne uygun filtre uygulayarak probleme ait özniteliklerin verimli bir şekilde elde edilmesine olanak sağlayabilir.
- Büyük veri setleri üzerinde analiz işlemi sonucunda karmaşık örüntüleri algılayabilir ve yüksek doğruluk değerleri elde edebilir.
- Görüntüler üzerindeki bozukluk ve gürültüye rağmen örüntüler doğru şekilde tanımlanabilir. Gürültüye karşılık dayanıklılığı vardır.
- Transfer öğrenimi desteği sunarlar. Böylece bir problemi çözmek için eğitilen bir ESA farklı bir görevi çok az veya hiç eğitim almadan başarılı ile tamamlayabilir.
- Öznitelik çıkarma işlemi otomatikleştirir. Görüntülerdeki örüntüler manuel işlem ihtiyacı olmadan ayırt edilebilir hale getirilir.

ESA'nın genel dezavantajları eğitim ve çalışma anı sırasında yüksek miktarda sistem kaynağına (hesaplama karmaşıklığı yüksek) gereksinim duymaları ve küçük veri kümeleri üzerinde aşırı ezberleme (overfitting) problemi yüzünden eğitim sırasında kullanılmayan veri üzerinde kötü tahminleme skoru elde etmesidir (Alzubaidi vd., 2021).

## 2.2.1. Evrişimli Sinir Ağı Katmanları

Basit ESA'lar üç katmandan oluşmaktadır. Bu katmanlar : Evrişim katmanı, havuzlama katmanı ve tam bağlı katmandır. Bu katmanların arka arkaya kullanılması ile ESA mimarisi oluşturulmaktadır. Beş katmanlı basit bir ESA mimarisi Şekil 4 ile gösterilmiştir.



Şekil 4. Beş katmanlı ESA mimarisi örneği (O'Shea ve Nash, 2015).

Bu örneğin temel işlevleri dört adıma ayrılabilir :

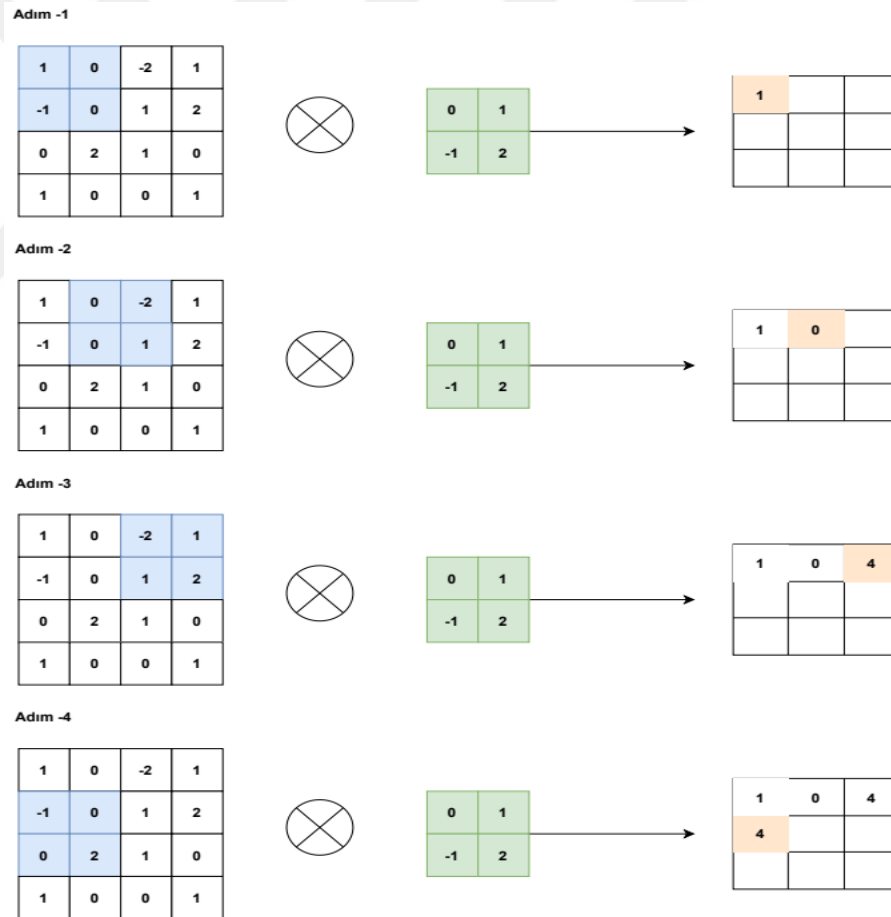
- 1) Giriş katmanı, görüntülere ait piksel değerlerini tutmaktadır.
- 2) Evrişim katmanı, girdinin yerel bölgelerine bağlı çıktı nöronlarını belirlemek için ağırlıklar ve girdi hacmine bağlı bölge arasındaki skalar çarpım hesaplanarak belirlenmektedir. Doğrultulmuş Lineer Birim (Rectified Linear Unit, ReLu) önceki katmandan üretilen aktivasyon çıktısına sigmoid benzeri bir aktivasyon fonksiyonu uygulamaktadır.
- 3) Havuzlama katmanında, girdinin uzamsal boyutunda alt örnekleme (downsampling) gerçekleştirilmektedir. Bu aşamadaki amaç parametre sayısını azaltmaktır.
- 4) Son olarak tam bağlı katmanlar, sınıflandırma için kullanılacak aktivasyon fonksiyonlarının puanlarını üretecektir. Performansı arttırmak için bu katmanlar arasında ReLu kullanılması önerilmektedir (O'Shea ve Nash, 2015).

### Evrişim Katmanı

ESA mimarisindeki en önemli bileşen evrişim katmanıdır. Bu katman, evrişimli filtre koleksiyonundan (kernel, çekirdek) oluşmaktadır. Girdi olarak verilen görüntü bu



filtreler ile birleştirilerek çıktı özellik haritası oluşturulur. Çekirdek, ayırık sayılar ve değerlerden oluşan bir matris olarak tanımlanabilir. Çekirdeğe ait her bir değere çekirdek ağırlığı denilmektedir. ESA eğitimi başlangıcı bu ağırlık değerleri rastgele olacak şekilde oluşturulmaktadır. Her eğitim adımı sonrası bu değerler tekrar ayarlanır ve çekirdeğin önemli öznelikleri çıkarsaması sağlanır. Evrişim işlemini yakında incelemek için 2x2'lik rastgele ağırlıkla başlatılmış çekirdeğe sahip olan 4x4'lük gri ölçekli bir görüntüyü örnek alalım. Şekil 5'te görülebileceği üzere çekirdek tüm görüntünün üzerinde yatay ve dikey olarak kaymakta bu sırada görüntüye ait giriş değerleri ile çekirdek arasındaki iç çarpım belirlenmektedir. Bu iç çarpım sırasında karşılık değerler çarpılmakta ve sonrasında toplanarak tek bir değer elde edilmektedir. Tüm süreç kayma imkanı kalmayana kadar devam ettirilir. Hesaplanan değerler, çıktıya ait özellik haritasını temsil etmektedir.



Şekil 5. Evrişim katmanında her adımda gerçekleşen hesaplamalar (Alzubaidi vd., 2021).

Şekil 5 üzerindeki açık mavi rengi çekirdek ile o adım için çarpılacak gerçek görüntü verisini, açık yeşil çekirdeğe ait 2x2'lik çekirdek ağırlıklarını ve açık turuncu ise çıktı özellik haritasına ait bir değeri temsil etmektedir.

ESA'nın önemli avantajlarından ikisi bu katman sayesinde oluşmaktadır. Bu avantajlardan ilki seyrek bağlantıdır. Tam bağlı katmanlarda her nöron bir sonraki katmandaki tüm nöronlar ile bağlantı kurmaktadır. ESA'larda ise iki bitişik katman arasında sadece birkaç ağırlık bulunmaktadır. ESA oluşturmak için gerekli olan ağırlık ve bağlantı sayısı daha azdır. Bu durum ağırlıkların depolanması için gerekli olan belleğinde düşük olmasını sağlamaktadır. ESA'ların başka bir avantajı ise ağırlık paylaşımıdır. ESA'larda komşu katmanlarda bulunan herhangi iki nöron arasında bir ağırlık bulunmamaktadır. ESA yapısından dolayı tüm girdi için tek bir ağırlık grubu öğrenilmesi gerekmektedir. Bu sayede her bir nöron için ağırlıkların öğrenilmesi gerekmediği için gerekli eğitim süresi ve çeşitli maliyetlerden büyük ölçüde kazanç sağlanmaktadır (Alzubaidi vd., 2021).

### **Havuzlama Katmanı**

Havuzlama katmanının amacı bir önceki katmanda oluşturulan özellik haritalarının alt örnekleme (sub-sampling) gerçekleşmesi işlemidir. Bu aşamada büyük boyutlu özellik haritaları küçültülerek, küçük boyutlu özellik haritaları oluşturulmaktadır. Bu katmanda kullanılmak amacıyla üretilmiş çeşitli havuzlama yöntemleri mevcuttur. Bunlara örnek olarak ortalama havuzlama, minimum havuzlama, maksimum havuzlama, genel ortalama havuzlama, genel maksimum havuzlama yöntemleri verilebilir (Alzubaidi vd., 2021). Tez kapsamında maksimum havuzlama, ortalama havuzlama, tamamen-evrimsel havuzlama (Springenberg vd., 2014), inception havuzlama (Szegedy vd., 2017) yöntemlerinden yararlanılmıştır.

### **Tam Bağlı Katman**

Genellikle ESA mimarilerinin son katmanı olarak kullanılmaktadır. Bu katmandaki her bir nöron bir önceki katmandaki tüm nöronlar ile bire bir bağlıdır. Bir tür ileri beslemeli yapay sinir ağı olduğu için geleneksel çok katmanlı algılayıcı sinir ağına benzemektedir. Bu katmana girdi ya son kullanılan evrimsel katmanı ya da son kullanılan

havuzlama katmanı üzerinden gelmektedir. Bu katmanın ürettiği çıktı ESA'nın nihai çıktısını oluşturmaktadır (Alzubaidi vd., 2021).

### **Sönümlenme Katmanı**

Sönümlenme (Dropout) katmanı (Srivastava vd., 2014), sinir ağlarının düzenlenmesi (regularization) için kullanılan en popüler yöntemdir (Park ve Kwak, 2017). Bu katmanda ağı düzenli hale getirmek için her adım sonrası aktivasyonların daha önceden belirlenmiş bir kısmı sönümlenir (bırakılır, unutulur). Genel bakış açısıyla farklı veri alt kümeleri kullanılarak eğitilmiş birden fazla modelin işbirliği olarak görülebilir. Özellikle tam bağlı katmanlar ile birlikte kullanılması durumunda aşırı ezberleme ile ilgili problemlerin çözümüne yardımcı olarak verimli sonuçlar elde edilmesini sağlamaktadır. Evrişim katmanlarından sonra kullanılabilir. Ancak evrişim katmanlarında aşırı ezberleme problemi yaşanmadığı ve tam bağlı katmana oranla parametre sayısı görece olarak az olduğu için bu katman sonra kullanımı genellikle tercih edilmemektedir. Özellikle evrişim katmanı sonrası yüksek sönümlenme olasılığı (0.5) kullanılmasının ESA'larda kötü performansa sebep olduğu görülmüştür (Park ve Kwak, 2017). Tez kapsamında ESA'ların üretim, aşamasında evrişim ve tam bağlı katmanlar sonrasında kullanılmak üzere sönümlenme katmanlarından yararlanılmıştır. Kullandığımız veri kümesi ve o anki ESA mimarisi için sönümlenme olasılığı değerinin hangi aralıklarda bulunması gerektiği araştırılmıştır.

### **Toplu Normalleştirme**

Toplu Normalleştirme (Batch Normalization, BN), yapay sinir ağlarının performansını ve kararlılığını geliştirmek için kullanılan bir tekniktir. Bu yöntem ile ağıdaki her bir katmanın girdileri normalleştirilerek eğitim sırasında gerçekleşme ihtimali olan dahili ortak değişken kayması (internal covariate shift) problemi ortadan kaldırılmaya çalışılmaktadır. Bu problem derin ağlarda eğitim sırasında gerçekleşmektedir. Problem kendilerinde önceki katmanların parametreleri değiştikçe, her katmana ait girdilerin dağılımının değişmesi gereksiniminden kaynaklanmaktadır. Her katmanın uyum sağlaması için gereken süre eğitim sürecini yavaşlatabilir. Toplu Normalleştirme, girdi değerlerini normalize ederek aktivasyon fonksiyonunun doyuma ulaşmasını engeller. Bu sayede ağların eğitim hızı ve doğruluğu artırılabilir. Daha yüksek öğrenme oranlarının

kullanılmasını sağlaması, başlangıç parametrelerinin iyi seçilmesi gerekliliğini ortadan kaldırması, aşırı ezberlemeyi azaltan bir düzeltici görevi görmesi gibi çeşitli faydaları vardır (Ioffe ve Szegedy, 2015). Toplu normalleştirme ile sönümleme işlemleri, ağ düzenlenmesi açısından benzer işlemi gerçekleştirdikleri için iki yöntemin birlikte kullanılması önerilmese de literatürdeki farklı araştırmalarda (Garbin vd., 2020) bu iki katman ve katmanların birlikte kullanımı karşılaştırılmıştır. Sonuçlar ESA'lar için BN'nin neredeyse her durumda test skoruna pozitif etki gösterdiğini ortaya koymuştur. Sonuçlara göre Sönümleme Katmanının sadece bazı durumlarda pozitif etkisi görülmüştür. Özellikle öğrenme oranının yüksek olduğu durumlarda (%0.001) Sönümleme Katmanı verimsiz sonuçlar elde etmiştir. Tez kapsamında oluşturulan hiperparametre uzayı sayesinde Toplu Normalleştirme ve Sönümle Katmanlarından birlikte yararlanma ihtimali bulunan ESA modelleri üretilmiştir. Hangi yöntemin görece olarak daha iyi sonuçlar elde ettiği araştırılmıştır.

### 2.3. Optimizasyon Algoritmaları

Bu kısımda literatürde parametrelerin optimizasyonu için sıklıkla kullanılan, tez kapsamında oluşturduğumuz modellerde yararlandığımız farklı optimizasyon algoritmaları hakkında bilgi verilecektir.

#### 2.3.1. Karelerin Ortalamasının Karekök Yayılımı

Karelerin Ortalamasının Karekök Yayılımı (Root Mean Squared Propagation, RMSP), yapay sinir ağlarının eğitimi için tasarlanmış bir optimizasyon algoritmasıdır. Stokastik Eğimli İniş algoritmasının bir uzantısıdır (Huang, 2020).  $W_t = t$  zamanında ağırlıklar,  $W_{t+1} = t + 1$  zamanında ağırlıklar,  $\alpha_t = t$  zamanında öğrenme oranı,  $\partial L =$  Kayıp fonksiyonunun türevleri,  $\partial W_t = t$  zamanında ağırlıkların türevi,  $V_t =$  Geçmiş gradyanların karelerinin toplamı (başlangıçta sıfır),  $\beta =$  Hareketli ortalama parametresi (sabit değer, 0,9),  $\epsilon =$  Küçük pozitif sabit sayı (sabit,  $10^{-8}$ ) olmak üzere bir sonraki zaman için ağırlıkların hesaplanması için öncelikle geçmiş gradyanların karelerinin toplamı ( $V_t$ ) Eşitlik 2.2 yardımıyla hesaplanmaktadır. Sonrasında elde edilen değer ( $V_t$ ) bir sonraki zamanda ( $W_{t+1}$ ) kullanılacak ağırlıkların hesaplanabilmesi için Eşitlik 2.1'de yerine konularak uygun ağırlıklar oluşturulmaktadır.

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t + \epsilon)^{1/2}} * \left[ \frac{\partial L}{\partial W_t} \right] \quad (2.1)$$

$$v_t = \beta v_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial W_t} \right]^2 \quad (2.2)$$

RMSP, salınımları dikey yönde kısıtlar. Bu nedenle, öğrenme oranını arttırılabilmekte veya algoritma yatay yönde daha büyük adımlar atabilir hale getirilebilmektedir.

### 2.3.2. Adaptif Moment Tahmini

Adaptif Moment Tahmini (Adaptive Moment Estimation, ADAM), stokastik amaç fonksiyonlarının optimizasyonu için birinci dereceden gradyan tabanlarını optimize etmek amacıyla düşük dereceli momentlerin uyarlanabilir bir şekilde tahminlenmesini sağlayan bir algoritmadır (Kingma ve Ba, 2017). Uygulanması basit ve hesaplama açısından verimlidir. Yöntemin uygulanması için düşük miktarda belleğe ihtiyaç duyulmaktadır. Büyük miktarda veri, parametre içeren problemlerin çözümü için uygundur. Genel yapısı Momentumlu eğimli iniş ve RMSP algoritmalarının birlikte kullanılması sayesinde oluşmaktadır (prakharr0y, 2020).

Adam algoritmasında kullanılan yöntemlerden biri olan momentum yöntemi, eğimli inişin daha hızlı bir şekilde minimuma ulaşmasını sağlamak için eğimlerin, üssel ağırlık ortalamasından (exponentially weighted average) yararlanılması sayesinde gerçekleştirilmektedir.  $m_t = t$  zamanı için gradyanların toplamı,  $m_{t-1} = t-1$  zamanı için gradyanların toplamı,  $W_t = t$  zamanı için ağırlıklar,  $W_{t+1} = t+1$  zamanı için ağırlıklar,  $\alpha_t = t$  zamanı için öğrenme oranı,  $\partial L =$  kayıp fonksiyonunun türevi,  $\partial W_t = t$  zamanındaki ağırlıkların türevi,  $\beta =$  Hareketli ortalama parametresi (sabit değer, 0,9) olmak üzere algoritma Eşitlik 2.3 ve Eşitlik 2.4 yardımı ile tanımlanabilir. Bu algoritmalarda öncelikle Eşitlik 2.4 yardımıyla  $t$  zamanı için gradyanların toplamı ( $m_t$ ) hesaplanmaktadır. Sonrasında elde edilen değer Eşitlik 2.3 yardımıyla bir sonraki zaman ( $t+1$ ) için ağırlıkların hesaplanması için kullanılmaktadır.

$$w_{t+1} = w_t - \alpha m_t \quad (2.3)$$

$$m_t = \beta m_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial W_t} \right] \quad (2.4)$$

ADAM algoritması, momentum yöntemi ve RMSP yönteminin birleştirilmesi sonucunda oluşmakta ve her iki yönteme oranla daha verimli bir eğimli iniş gerçekleştirmektedir. ADAM algoritmasında Eşitlik 2.4 ve Eşitlik 2.2'de verilen eşitliklerin birlikte kullanılabilmesi için hareketli ortalama parametresi ( $\beta$ ) iki farklı değer olacak şekilde yeniden oluşturulmaktadır.  $\beta_1 = 0,9$  ve  $\beta_2 = 0,999$  olmak üzere oluşturulan eşitliklerin yeni versiyonları Eşitlik 2.5 ve Eşitlik 2.6 ile gösterilmektedir.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[ \frac{\partial L}{\partial W_t} \right] \quad (2.5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[ \frac{\partial L}{\partial W_t} \right]^2 \quad (2.6)$$

Algoritma ile gerçekleştirilen başka bir iyileştirme ise  $m_t$  ve  $v_t$  değerleri üzerinedir. Başlangıçta 0 varsayılan değerine sahip olan bu değerler  $\beta_1$  ve  $\beta_2$  değerleri 1'e yakınsadığı için 0 doğru sapma eğilimi göstermektedirler. Bu eğilimi düzeltmek için ADAM algoritmasında sapma eğilimi düzeltilmiş  $m_t$  ve  $v_t$  değerleri hesaplanmaktadır. Sapma eğiliminin düzeltilmesi küresel minimuma ulaşırken ağırlıkları kontrol etmek ve küresel minimum yaklaşırken yüksek salınımları önleme konusunda avantaj sağlamaktadır. Sapma eğilimi düzeltilmiş değerlerin hesaplanması için kullanılan eşitlikler Eşitlik 2.7 ve Eşitlik 2.8 ile gösterilmiştir.

$$\hat{m}_t = \frac{m_t}{(1 - \beta_1^t)} \quad (2.7)$$

$$\hat{v}_t = \frac{v_t}{(1 - \beta_2^t)} \quad (2.8)$$

Eşitliklerin birleştirilmesi ile ADAM algoritmasının gerçekleyen Eşitlik 2.9 elde edilmektedir.

$$w_{t+1} = w_t - \hat{m}_t \left( \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \right) \quad (2.9)$$

Bu eşitlikteki  $\alpha$ , öğrenme oranı (0,001) ve  $\epsilon$ , sıfıra bölme hatasını engellemek için kullanılan küçük ( $10^{-8}$ ) bir sabit değerdir.

### 2.3.3. Düzeltilmiş Adaptif Moment Tahmini

Düzeltilmiş Adaptif Moment Tahmini (Rectified Adaptive Moment Estimation, RADAM), ADAM optimizasyonunun bir varyasyonudur. RADAM'ın genel amacı ADAM'da bulunan uyarlanabilir öğrenme oranına ait varyansı düzeltmektir (Liu vd., 2019) RADAM yazarları, ADAM'ın özellikle model eğitiminin erken aşamalarında, eğitim için kullanılan örneklerin sınırlı bir kısmından yararlanılmış olmasına rağmen öğrenme oranında gerçekleşebilen büyük varyansın global minimuma yakınsama konusunda olumsuz etki gösterdiğini savunmuştur. RADAM'da bu etkiyi azaltabilmek için öğrenme oranını düzelten bir terimden yararlanmaktadır.

RADAM ile ilk 4 iterasyona ıraksak (divergent) aşama denilmektedir. Bu aşamada ağır-top momentum stokastik eğimli iniş gerçekleştirilmektedir. 5. iterasyon ve sonrasında ise algoritma yakınsak (convergent) aşamaya geçiş yapmaktadır. Bu aşamada ise sabit ısınmaya programına (fixed warmup schedule) sahip ADAM algoritmasından yararlanılmaktadır (Ma ve Yarats, 2021).

### 2.3.4. Nesterov ile Hızlandırılmış Uyarlanabilir Moment Tahmini

Nesterov ile Hızlandırılmış Uyarlanabilir Moment Tahmini (Nesterov-accelerated Adaptive Moment Estimation, NADAM), sinir ağı modellerinde kullanılan bir optimize edicidir. Yöntem, güncelleme sırasında maliyet fonksiyonunu en aza indirme amacıyla ağırlıklar için optimum değerleri bulmaya çalışmaktadır. Genellikle gürültü eğimler ve

eğriliğin yüksek olması durumunda kullanılmaktadır (Greis vd., 2020). ADAM optimizasyonunun bir başka varyasyonudur. NADAM, teorik açıdan ADAM'da kullanılan momentumdan üstün olan Nesterov momentumundan yararlanmaktadır. Bu özelliğinden ötürü çoğu durumda ADAM'a oranla daha verimli bir şekilde optimizasyon gerçekleştirdiği görülmüştür (Dozat, 2016).

#### **2.4. Mısır Tohumu Sınıflandırılması**

Bu bölümde mısır tohumlarının sınıflandırılması için kullanılan yöntemler ile ilgili çalışmalardan bazıları incelenmiş özellikle makine öğrenmesi tabanlı yöntemler detaylı bir şekilde incelenerek bu yöntemlerin eksi ve artıları belirlenmeye çalışılmıştır. Bu bölümde ayrıca mısırdan bağımsız olarak ESA'larda hiperparametre optimizasyonu gerçekleştirmeye çalışan birkaç araştırmada yer almaktadır. Literatürde mısır tohumlarının sınıflandırılması problemini çözmeye çalışan bir çok araştırma bulunmaktadır. Gerçekleştirilen çalışmalarda genel yöntem bir sensör yardımı ile tohum hakkında bilgi elde etmek, sonrasında bu bilgiden yararlanarak sınıflandırmanın gerçekleştirilebilmesi için bir örüntü yakalamaktır. Örüntünün bulunması için günümüzdeki çalışmaların büyük çoğunluğunda makine öğrenmesi yöntemlerinden yararlanılmıştır. Literatürdeki önemli çalışmalardan biri 2013'te (Couto vd., 2013) gerçekleştirilmiştir. Bu çalışma ile R1-nj işaretleyicisinin bazı tohumlar için ifade gücünün yeterli olmadığından dolayı tohumlarda oluşan rengin tonlarının yetersiz olduğu öne sürülmüş, rengin embryo yada endospermde benzer olabildiğini göz önünde bulundurarak tohumların biçimlerine ve tohumlara ait ölçülere bakılarak sınıflandırma işleminin gerçekleştirilip gerçekleştirilemeyeceği araştırılmıştır. R1-nj işaretleyicisinin renk değişimine sebep olmadığı 397 tohum (Haploid olduğu kabul edilen tohumlar, Putative Haploid) ağırlık, genişlik, uzunluk, kalınlık olarak ölçülmüş ve çimlendirilmiştir. Akış sitometresi kullanılarak dört farklı haploid tohumu taranmıştır. Bu sonuçlardan yararlanılarak tohumlara karşılık gelen ölçü ve biçimler dikkate alınarak örüntü ilişkilendirilmesi yapılmıştır. Sonuç olarak ağırlık, uzunluk, kalınlık, genişlik gibi özelliklerin haploid tohumların sınıflandırılması için uygun olmadığı bulunmuştur. Çimlendirilmiş tohumlar içinse haploid olduğu kanıtlanmış dört bitkinin diploidlere oranla daha küçük boyuta, daha kırılğan yapraklara sahip olduğu ve diploidlere oranla daha farklı bir yaprak düzenine sahip olduğu gözlenmiştir. Bu çalışma ile mısır tohumlarının sadece fiziksel özelliklerine (ağırlık, kalınlık, uzunluk, genişlik) bakılarak



haploid yada diploid olarak sınıflandırılmayacağı kanıtlanmıştır. Çalışma ile R1-nj işaretleyicisinin tohum aşamasında sınıflandırma yapılabilmesi için önemli bir faktör olduğuna dikkat çekilmektedir. Ayrıca çimlenmiş tohumlarda fiziksel özelliklere (haploidlerde kırılğan yaprak, daha küçük bitki, diploidlere göre yaprak düzeni farkı) bakılarak ayırım yapılabileceği öne sürülmüştür. Tohumların çimlendirildikten sonra ayırımlarının yapılmasının zaman ve iş gücü açısından ekstra maliyet oluşturacağı göz önünde bulundurulmalıdır. İndüklenmiş mısır tohumlarının sadece %2 ile %10 arasında bir miktarının haploid olacağı göz önünde alındığında tohumların çimlendirilmeden sınıflandırılmasının önemi ortaya çıkmaktadır. Literatürdeki çalışmaların büyük kısmı bu yüzden tohumları çimlendirmeden önce sınıflandırmaya çalışmaktadır.

Boote vd. (2016) tarafından gerçekleştirilen bir araştırmada, floresan emisyonunun, soğurma bazlı optik sinyallere (renk) oranla genellikle daha seçici olduğunu öne sürülerek mısır tohumlarının sınıflandırılması için floresan spektroskopisinden yararlanmıştır. Üç farklı öznelik üzerinden sınıflandırma işlemi gerçekleştirilmiştir. Bunlar sırasıyla floresan yoğunluğu (yayılan foton sayısı), dalga boyu(yayılan foton enerjisi) ve mekansal bağımlılıktır (çekirdek boyunca lokalize olmuş varyasyon). Çalışma kapsamında yedi kendilenmiş hat analiz edilmiştir. Bu analizin sonucunda her hattın kendine göre çekirdek, embriyo tipine ve boya lekesine sahip olmasından dolayı verimli sınıflandırma yapılabilmesi için floresan yoğunluğu ve alan eşiği değerlerinin her hat için optimize edilmesi gerektiği görülmüştür. Yapılan sınıflandırma işlemi sonrası sınıflandırmanın doğruluğu, tohumlardan üretilen bitkilere DNA işaretleyici analizi uygulanması ile kanıtlanmıştır. Sonuç olarak kullanılan hatta bağlı olarak %80-%90 aralığında sınıflandırma verimliliğine sahip bu yöntemin tohum başına 3,3 saniye gibi bir sürede 2400 tohumu (12 dakika) analiz edebileceği öne sürülmüştür. Bu araştırmada öne sürülen yöntem kullanılan sensörün ulaşılabilirliği ve maliyeti, farklı hatlar için yönteme ait parametrelerde değişikliğe ihtiyaç duyulduğu için evrensellik ve tekrarlanabilirlik açısından sınırlı kapasiteye sahiptir.

Altuntaş vd. (2018) tarafından yürütülen bir araştırma ile 87 haploid 326 diploid olan 413 adet RGB tohum görüntüsünden yararlanarak 5 farklı öznelik çıkarımı yapmıştır. Tohum görüntülerinin orta kısmında embriyoyu temsil eden 35'er piksellik çember kesitler alınmış, çemberin içinde kalan kısımlar embriyo kabul edilmiştir. Bu kesite

ait R,G,B renk değerlerinin ortalama değerleri birer öznitelik olarak atanmıştır. Bu ortalama değerlerin bulunması için kesitin içinde kalan R,G,B değerlerinin her biri toplanarak toplam piksel sayısına bölünmüştür. Diğer iki özneliğin hesaplanması için R1-nj renklenmesinin embriyo ve endosperm üzerinde gerçekleşip gerçekleşmediği araştırılmıştır. Örnekler gri seviyeli görüntüye çevrilmiş sonrasında deneyler sonucu belirlenen 100 eşik değeri kullanılarak ikili görüntüye dönüştürülmüş ve son olarak terslenmiştir. İkili görüntüde bulunan 1 değerine sahip her piksel R1-nj renklenmesine sahiptir kabulü yapılmıştır. Embriyo ve endosperm bölümleri için ayrı olacak şekilde piksel toplamları hesaplanmış ve piksel değerleri toplamı 0'dan büyükse renklenme var, değilse renklenme yok denilmiştir. Elde edilen niteliklerden yararlanılarak Destek Vektörü Makineleri (Support Vector Machines, SVM) kullanarak sınıflandırma gerçekleştirmiştir. Yöntemin haploid tohumları %94,25, diploid tohumları %77,91 doğruluk ile sınıflandırabildiği görülmüştür. Veri seti için ortalama başarı oranı %81,36 olarak bulunmuştur. Bu çalışmada kullanılan RGB görüntüler, floresan spektroskopisi gibi pahalı ve karmaşık ekipmanlara ihtiyaç duyulmadan elde edilebilmektedir. RGB görüntüler, tohumun renk, doku ve morfoloji gibi farklı yönlerini tanımlayarak ayırt edici özelliklere vurgu yapabilir. Bu durum pozitif yada negatif yönlü etki oluşturabilir. Çalışmada gerçekleştirilen ikili görüntüye çevirme işlemi için kullanılan 100 eşik değeri çalışmanın tekrarlanabilirliğini büyük ölçüde düşürmüştür. RGB görüntülerin çekimi sırasında ortamdaki aydınlatma miktarına bağlı olarak bu eşik değerinin değişmesi gerektiği göze alındığında yöntemin farklı veri kümeleri üzerinde verimli bir şekilde çalışması beklenemez.

Veeramani vd. (2018) tarafından gerçekleştirilen bir çalışmada, araştırmacılar kendi oluşturdukları 3779 diploid, 952 haploid RGB tohumdan görüntüsünden oluşan veri kümesinden yararlanarak tohum sınıflandırma işlemini gerçekleştirmiştir. Sınıflandırma işlemi için kendi ürettikleri ESA bazlı DeepSort modelinin doğruluk değerleri ile SVM, Rastgele Orman Algoritması (Random Forest Algorithm, RFA) ve Lojistik Regresyon (Logistic Regression, LR) yöntemlerinin doğruluk değerleri karşılaştırmıştır. Elde edilen doğruluk değerleri DeepSort için %98,6, SVM için %97,6, RFA için %84,5, LR için %77,5 olarak bulunmuştur. Bu çalışmada her ne kadar yüksek başarı oranı elde etmiş olsa da kullanılan veri setinde sınıfların örnek sayılarının arasında bulunan büyük farklılıktan (3779 diploid, 952 haploid) ötürü sınıf dengesizliği problemine sahiptir. Sınıf dengesizliği, eğitim veri seti

içerisinde bulunan sınıfların örnek sayılarının birbirinden çok farklı olması durumudur. Bu durum özellikle azınlık sınıf (haploid) için kötü tahminleme performansına yol açmaktadır. Sınıf dengesizliği probleminin negatif etkilerini azaltan Undersampling, Oversampling yöntemlerinden yararlanılmamıştır. Araştırmada, sınıf dengesizliği bulunan durumlarda, modellerin performansını karşılaştırmak için kullanılması daha doğru olan hassasiyet, geri çağırma, F1 skoru gibi metriklere ait değerler paylaşılmamıştır.

L. Yu vd. (2018) tarafından yürütülen bir araştırmada, yakın kızılötesi (Near Infrared, NIR) spektrumdan yararlanarak tohumların sınıflandırma işlemini gerçekleştirmiştir. Bu teknolojinin kullanılması sayesinde tohumlara zarar verilmeden, hızlı ve düşük maliyetli bir şekilde sınıflandırmanın gerçekleştirilebileceği öne sürülmüştür. Çalışmada ışığa, sıcaklığa, ortamdaki kızıl ötesi ışın yoğunluğuna, kızıl ötesi veri alan ekipmana ve tohum aktivesinin dinamik değişimine göre tohumların yakın kızılötesi spektrumları doğrusal olmayan özellikler gösterdiği söylenmiştir. Bu yüzden çalışma kapsamında doğrusal olmayan öznitelik analitik yöntemi olan Denetimli Sanal Örnek Çekirdek Yeri Koruma Projeksiyonu (Supervised Virtual Sample Kernel Locality Preserving Projection , SVSKLPP) kullanılmıştır. Bu yöntem öznitelik çıkarımı için kullanılmış veri sonrasında SVM kullanılarak sınıflandırılmıştır. Veri seti olarak yakın kızılötesi spektrum ölçümü yapan cihaz kullanılarak farklı zamanlarda 100 haploid, 100 hibrit tohumdan içeren eğitim seti (T1) ve 100 haploid, 100 hibrit tohum içeren test veri seti (T2) oluşturulmuştur. SVSKLPP yöntemi için ortalama doğruluk (accuracy) %97,1 , duyarlılık (sensitivity) %98,8 , özgüllük (specificity) %95,4 değerleri elde edilmiştir. Yakın kızıl ötesi spektrum tabanlı başka bir çalışmada 2018'de (Lin vd., 2018) gerçekleştirilmiştir. Araştırma, mısırlara ait NIR spektrumlarının, tanımlama performansını düşürecek birçok gereksiz özellik ve gürültü içerdiğini öne sürmüştür. Gereksiz özellik ve gürültünün oluşturduğu negatif etkiyi indirmek için bu çalışmada özellik uzayını tekdüze (uniform) hale getirmek için Sıfır Düzeyli Bileşen Analizi (Zero-phase Component Analysis, ZCA) yönteminden ve düşük boyutlu uzay tanımlaması için Kısmi En Küçük Kareler Regresyonu (Partial Least Squares Estimation, PLSE) yönteminden yararlanılmıştır. Veri seti olarak kullanılan 250 haploid ve 250 hibrit tohumun 200'ü eğitim (%40), 100'ü validasyon (%20), 200'ü test (%40) için ayrılmıştır. Sonrasında Geri Yayılımlı Sinir Ağı Sınıflandırıcısı (Back Propagation Neural Network, BPNN) yardımıyla sınıflandırma işlemi gerçekleştirilmiştir. Yöntemin ortalama doğruluğu %96,16 olarak bulunmuştur.

Araştırmalardan görülebileceği üzere NIR bazlı çalışmaların genel problemi gürültü ve gereksiz özelliklerin elemine edilmesi için yöntem, yöntemler grubu uygulanması ihtiyacıdır. NIR spektrumda ölçüm yapabilen sensörler, RGB bazlı sensörlere oranla erişim olanağı ve maliyet konularında dezavantajlı pozisyonundadır.

Wang vd. (2018) tarafından gerçekleştirilen bir araştırmada yakın kızılötesi hiperspektrum görüntüleme teknolojilerinden yararlanılarak haploid mısır çekirdeğini hızlı ve doğru bir şekilde tanımlaması amaçlanmıştır. Araştırma, embriyosu gözükken ve gözükmeyen (tohum sırtı dönükken çekim yapılmış) haploid ve diploid mısır tohumları arasında yapılan ölçümlerde farkın daha büyük olduğunu ancak yine de etkili bir şekilde ayırmanın yapılabildiğini öne sürmüştür. Bu yönden diğer yöntemlere oranla büyük bir avantaj sağlamaktadır. Araştırmada 100 adet diploid, 100 adet haploid tohuma ait spektral verilerden yararlanılmıştır. Eğitim ve test seti sırasıyla ikiye bir oranda (%66,%33) olacak şekilde ayarlanmıştır. Eğitim seti içerisinde bulunan anormal örnekler elimine edilmiş Biyomimetik İlişkisiz Yerel Ayrımcı Projeksiyon (Biomimetic Uncorrelated Locality Discriminant Projection, BULDP) kullanılarak spektral nitelikler çıkarılmıştır. Biyomimetik Örüntü Tanıma (BPR, Biomimetic Pattern Recognition)'dan yararlanılarak model oluşturulmuştur. Araştırma sonuç olarak nitel analiz yöntemini kullanarak benzer yağ içeriğine sahip haploid ve diploid tohumlar için %99 Doğru Kabul Oranı (Correct Acceptance Rate, COR) , %1 altında Yanlış Kabul Oranı (False Acceptance Rate, FAR) elde etmiştir. Çalışma hiperspektral görüntüleme teknolojisinin haploid tohumların sınıflandırılması konusunda kullanılabileceğini göstermiştir. Hiperspektrum görüntüleme teknolojisinin, RGB ve NIR'dan en büyük farkı tohum sırtından alınmış verilerden yararlanılarak da verimli sonuçlar üretebilmesidir. Diğer görüntüleme yöntemlerinde sensörler veri almadan önce tohumların embriyo yukarı bakacak şekilde iki boyutlu düzlemde uygun şekilde yerleştirilme zorunluluğu bulunmaktadır. Hiperspektrum görüntüleme ile bu gereksinim ortadan kaldırılmıştır. Bu yöntemin genel dezavantajı kullanılan sensörlerin maliyetinin yüksek, sensörlere erişilebilirliğin düşük olmasıdır.

Taylor vd. (2019) tarafından üretilen bir araştırmada, mısır tohumlarını terahertz zaman alanı spektroskopisi ile elde ettikleri verilerden yararlanarak, Olasılıksal Sinir Ağı (Probabilistic Neural Network, PNN) makine öğrenmesi yönteminin yardımıyla haploid ve diploid olarak sınıflandırmıştır. Işın odağı ve çekirdeklerin morfolojik farklılıklarından

kaynaklanan problemleri minimize etmek için ölçümler farklı yüksekliklerden tekrarlanmıştır. Elde edilen dalga formu verileri frekans alanına dönüştürülmüş ve rastgele alt örnekleme (random subsampling) yöntemi kullanarak eğitim seti oluşturulmuştur. Bu eğitim seti PNN'den yararlanılarak eğitilmiştir. Tüm veri seti 44 diploid, 47 haploid olmak üzere toplamda 91 adet tohum içermektedir. Yöntem 5 kat çarpaz doğrulma kullanıldığında ortalama %75 validasyon doğruluğu elde etmiştir. Yöntem, ölçümlerin şartlara bağlı olarak farklı yüksekliklerden gerçekleştirilmesini gerektirdiği için tekrarlanabilirlik, farklı tohum genotiplerine adapte edilebilirlik açısından dezavantajlara sahiptir.

Bu kısma kadar yapılan çalışmalarda kullanılan veri kümeleri paylaşılmamış ve çalışmaların büyük çoğunluğunda görece olarak düşük sayıda sınıf örneklerinden (100-400 aralığı) yararlanılarak modeller eğitilmiştir. Altuntaş vd. (2019) tarafından gerçekleştirilen araştırma, hem kullandıkları veri kümesini paylaşmaları hemde görece olarak yüksek sayıda sınıf örneği (1230 haploid, 1770 diploid) kullanmalarından dolayı diğer araştırmalardan pozitif yönde ayrılmıştır. Bu çalışma ile ESA bazlı transfer öğrenme yaklaşımından yararlanılarak haploid ve diploid mısır tohumlarını hızlı ve düşük maliyetle sınıflandırmak için bir model üretilmeye çalışılmıştır. Önceden eğitilmiş Alexnet, VGG-16, VGG-19, GoogleNet, ResNet14, ResNet-50, ResNet-101 modellerinden yararlanılarak modeller arası karşılaştırma yapılmıştır. Çalışmada önceden eğitilmiş modeller üzerinde ince ayar (fine tuning) işlemi gerçekleştirilmiştir. En yüksek sınıflandırma doğruluğunu %94,22 ile VGG-19 modeli üretmiştir. VGG19 modeli %94,58 duyarlılık, %93,97 özgüllük, %94,27 kalite indeksi, %93,07 F1 skoru değerlerini elde etmiştir. Modelin eğitilebilir parametre sayısı 144 milyondur (Simonyan ve Zisserman, 2014). Tez çalışmasında bu veri kümesinden yararlanılmıştır.

Başka bir araştırmada (Dönmez, 2020), üstte verilen veri setinden (Altuntaş, Cömert vd., 2019) yararlanılarak GoogleNet, ResNet-18, ResNet-50 ve VGG-16'yı öznitelik çıkarımı için kullanmış sonrasında (k-NN), Karar Ağaçları (Decision Trees, DT) ve SVM'den yararlanarak tohumların sınıflandırma işlemi gerçekleştirilmiştir. En yüksek doğruluğa ResNet50 ile öznitelik çıkarımı sonrasında yapılan sınıflandırma işlemleri sırasında ulaşılmıştır. k-NN için %87,5, SVM için %91,4, DT için %82,5 doğruluk sonucu elde edilmiştir.

Haploide ayırma yöntemlerinin tamamı R1-nj yönteminden yararlanmamaktadır. Yakın geçmişte yapılan çalışmalar ile bu ayırmanın tohumların yağ oranına (Melchinger, Schipprack, Würschum vd., 2013) (Melchinger, Schipprack, Mi vd., 2015) bakılarak gerçekleştirilebileceği görülmüştür. Ancak bu sınıflandırma işleminin uygulanabilmesi için tohumların yüksek yağ indükleyiciler kullanılması gerekliliğini ortaya çıkarmaktadır. Özellikle R1-nj işaretleyicisinin kullanılması olanağı bulunmayan C1 inhibitör genine sahip tohum soylarında bu yöntemin kullanılması avantaj sağlamaktadır (Ford, 2000).

## 2.5. Hiperparametre Optimasyonu

Tezde ele alınan hiperparametre optimizasyonuna ait literatürdeki çalışmalar incelendiğinde (Sanders ve Giraud-Carrier, 2017) bu çalışmaların iki farklı varsayım üzerine gerçekleştirildiği görülmektedir. Bu varsayımlardan ilki hiperparametre ayarları önemli değildir. Varsayılan parametrelerin arasından seçim yapmak yeterlidir. İkinci varsayım ise hiperparametre değerleri performans üzerinde önemli bir etkiye sahip olabilir. Dolayısıyla her zaman optimize edilmeye çalışılmalıdır. Bu iki karşıt varsayımdan hangisinin doğru olduğu daha önce araştırma (Sanders ve Giraud-Carrier, 2017) konusu olmuştur. Yapılan çalışmada 229 farklı veri kümesinden yararlanarak genetik algoritma kullanarak SVM, Çok Katmanlı Algılayıcı (Multilayer Perceptron, MLP) ve DT için hiperparametre optimizasyonu yapılmıştır. Her veri kümesi algoritma kombinasyonu için parametre uzayından rastgele değerler kullanılarak 30 kez (30 kez x 3 algoritma x 229 veri kümesi = 20610 kez) genetik algoritma kullanılarak hiperparametre optimizasyonu gerçekleştirilmiştir. Gerçekleştirilen çalışma sonucu varsayılan parametrelerin tam doğru sonuç elde ettiği durumlar hariç her durumda hiperparametre optimizasyonu yapılması sonrasında modellerin sonuçlarında iyi yönde gelişme gerçekleştiği görülmüştür. Karar ağaçlarının varsayılan parametrelerin performansını geçmek için SVM ve MLP'ye göre daha düşük hesaplama maliyetine sahip olduğu bulunmuştur.

Hiperparametreler, ağın mimarisini belirlemek, eğitim yapılacak ağın özelliklerini atamak yada her ikisini birden gerçekleştirmek (Aszemi ve Dominic, 2019) amacıyla kullanılabilir. Örnek olarak 2017'de yapılan bir çalışma (Bochinski vd., 2017) ile ESA için optimum ağ yapısının bulunması için ağ yapısını oluşturan etmenler

hiperparametre olarak tanımlanmış ve hiperparametreler evrimsel algoritma kullanılarak optimize edilmeye çalışılmıştır.

Tez çalışmasında hibrit yaklaşım kullanılarak hem ağın özellikleri, hemde ağın mimarisi tanımlanan hiperparametreler yardımıyla eniyilenmiştir. Ağın mimarisini oluşturmak için kullanılan hiperparametreler: Evrişim çekirdek boyutu ne olmalı (convolutional kernel size), artık evrişim çekirdek boyutu (residual convolutional kernel size) ne olmalı, aktivasyon fonksiyonu olarak ne kullanılmalı, optimizasyon algoritması türü ne olmalı, evrişimli katman sönümlenme olasılığı ne kadar olmalı, tam bağlı katman sönümlenme olasılığı ne kadar olmalı, kümeleme normalizasyonu kullanılmalı mı, ilk katmanlar için spesifik evrişim katman mimarisi kullanılmalı mı, artık ağ kullanılmalı mı, evrişimli ağ gizli katman çarpanı değeri ne olmalı, evrişimli katmanlar arka arkaya maksimum kaç tane kullanılmalı, havuzlama tipi ne olmalı, tam bağlı katman nöron çarpanı ne olmalı, ikinci tam bağlı katman kullanılmalı mı sorularına cevap arayan hiperparametrelerinden oluşmaktadır. Eğitim yapılacak ağın özelliklerini tanımlamak için kullanılan hiperparametreler: Öğrenme katsayısı çarpanı (Learning rate multiplier) ne olmalı, öğrenme ağırlığı kaybı (Learning weight decay) ne olmalı, küme büyüklüğü (Batch size) ne olmalı sorularına cevap arayan hiperparametrelerden oluşmaktadır.

## ÜÇÜNCÜ BÖLÜM

### MATERYAL VE YÖNTEM

ESA'ların oluşturulması sırasında katmanların sırasının belirlenmesi, kullanılacak katmanlara karar verilmesi (Basha vd., 2020), katmanlara ve modele ait hiperparametrelerin değerlerinin belirlenmesi (Garbin vd., 2020) günümüzde hala güncel olan araştırma konularıdır. Bu bölümde bu konuların üzerinde durduğu sorulara cevap aramak için tez çalışması kapsamında geliştirilen melez sistemin geliştirimi için kullanılan yazılım ve donanım altyapısının detayları anlatılmaktadır. Yararlanılan veri kümesi kısaca anlatılmıştır. Ayrıca hiperparametre uzayını tanımlamak için kullanılan hyperopt kütüphanesinin parametre ifadeleri hakkında bilgi verilmiş ve bu ifadelerden yararlanarak tanımladığımız hiperparametre uzayı tanıtılmıştır. Hiperparametre optimizasyonu yapmak için kullandığımız Ağaç Bazlı Parzen Tahminleyicileri algoritması anlatılmış ve sonrasında modellerin üretim adımları, başlangıç parametreleri, ön işleme aşaması ve eğitim prosedürüne dair bilgiler verilmiştir.

#### 3.1. Çalışmada Kullanılan Yazılım ve Donanım Altyapısı

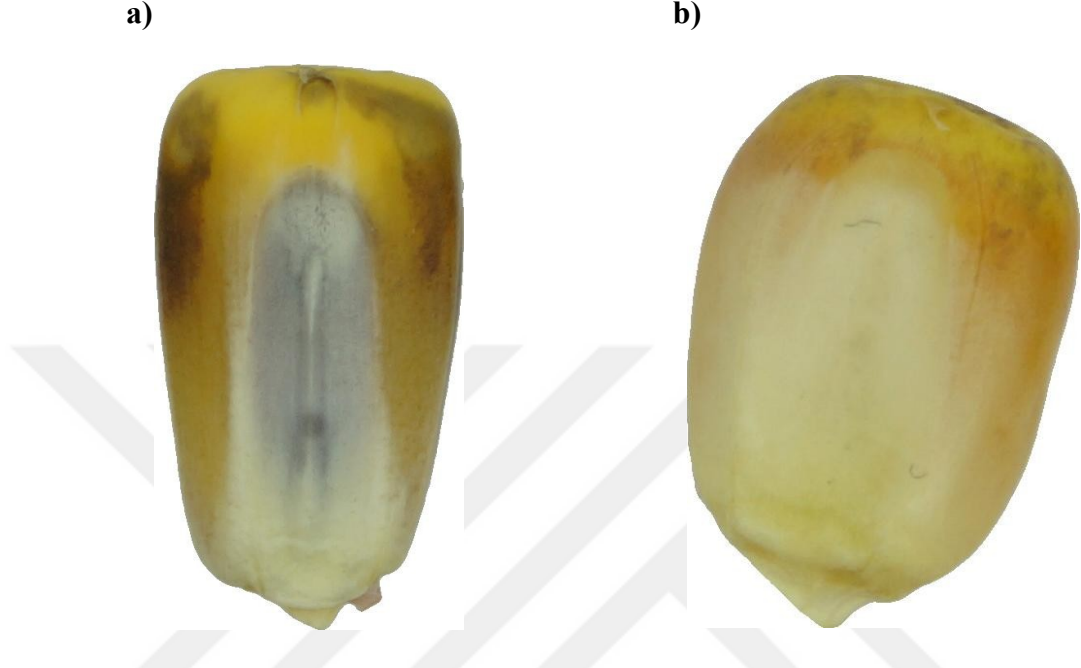
Bu çalışmada modellerin üretilmesi için Python programlama dili kullanılmıştır. Modellerin oluşturulması için kullanılan yapı Github'da bulunan ve Cifar-100 (guillaumechevalier, 2015) veri kümesi üzerinde parametre optimizasyonu gerçekleştiren kodun güncellenmesi ile elde edilmiştir. Kodların yazılması ve test edilmesi aşamasında Visual Studio Code tümleşik geliştirme ortamından yararlanılmıştır. Geliştirme için kullanılan kütüphane versiyonları Hyperopt için 0.2.7, Tensorflow için 2.10.1, Keras için 2.10, Numpy için 1.23.5'dir. Yazılım ve deneyler i9 Intel 12900 2.40 Ghz işlemci, GTX 3070 8 GB dahili ekran kartı, 64 GB RAM bellekli bilgisayar üzerinde gerçekleştirilmiştir.

#### 3.2. Veri Kümesi

Çalışma kapsamında hazır veri kümesi (Altuntaş, Cömert vd., 2019) kullanılmıştır. Bu veri kümesi R1-nj renk işaretleyicisine sahip haploid ve diploid mısır tohumları görüntülerinden oluşmaktadır. Veri kümesinde, 1230 haploid ve 1770 diploid olmak üzere 3000 mısır tohumu görüntüsü bulunmaktadır. Veri kümesini oluşturanlar tarafından



tohumlar R1- nj renk işaretleyicisinin farklı ifadeleri (az belirgin, çok belirgin gibi) yansıtılacak şekilde seçilmiştir. Veri kümesindeki görüntüleri çözünürlükleri, tohum boyutuna göre 300x289 ile 610x637 piksel arasında değişmektedir.



Şekil 6. (a) Diploid mısır tohumu örneği, (b) Haploid mısır tohumu örneği (Altuntaş vd., 2019)

Veri kümesinde bulunan haploid ve diploid görüntülere ait örnek Şekil 6'da görülebilir. Tohumun embriyo kısmında bulunan yukarı yönlü çizgi şeklinde, mor rengin tonlarında bulunan kısmın olup olmaması tohumun diploid mi ? Yoksa haploid mi ? Olduğunun uzmanlar tarafından belirlenmesinde kullanılmaktadır. Veri kümesinin incelenmesi sırasında bazı örneklerin sınıflandırma için uygun olmadığı görülmüş ancak diğer çalışmalarla (Altuntaş, Cömert vd, 2019) (Dönmez, 2020) ile karşılaştırma imkanı kaybetmemek için uygunsuz örnekler veri kümesinden çıkartılmamıştır. Örneklerin bazılarının hastalıklı olduğu, bazılarının ise embriyo bölgesinin zarar görmüş olduğu tespit edilmiştir. Bu istenmeyen durumlara sahip görüntüler, sınıflandırma işlemini zora sokmakta sahip olunabilecek maksimum sınıflandırma doğruluğunu düşürmektedir.

### 3.3. Hyperopt Stokastik Parametre İfadeleri

Hyperopt'un optimizasyon algoritmaları tarafından tanınan stokastik ifadeler (hyperopt, 2023) şunlardır: hp.choice(), hp.randint(), hp.uniform(), hp.quniform(),

hp.loguniform(), hp.qloguniform(), hp.normal(), hp.qnormal(), hp.lognormal(), hp.qlognormal(). Tez kapsamında bu ifadelerden: hp.loguniform(), hp.choice(), hp.uniform() ve hp.quniform()'dan yararlanılmıştır. Bu kısımda yararlanılan ifadeler hakkında detaylı bilgi verilecektir.

**hp.choice(etiket, seçenekler):** Bu yöntem etiket, ve seçenekler olmak üzere üzere iki girdiden yararlanmaktadır. Liste yada tuple olması gereken seçenekler değişkeni içerisinde bir elemanı döndürür. Seçeneklerin öğeleri stokastik olabilir. Bu durumda stokastik seçimler koşullu parametreler haline gelmektedir.

Örnek: hp.choice('optimizer', ['Adam', 'Nadam', 'RMSprop']) için optimizasyon yöntemlerinden rastgele birini döndürür.

optimizer tanım aralığı = 'Adam', 'Nadam', 'RMSprop'

**hp.uniform(etiket, alt, üst):** Bu fonksiyon etiket, alt ve üst olmak üzere üç girdiden yararlanmaktadır. Fonksiyon, alt ve üst değerler arasında homojen bir şekilde olmak üzere rastgele değer döndürmektedir. Optimizasyon işlemleri sırasında bu değişken iki taraflı aralığa sınırlandırılmaktadır.

Örnek: hp.uniform(öğrenme oranı, 0,0001, 0,001) için [0,0001, 0,001] aralığında rastgele değer döndürülmektedir. Fonksiyonun ilk çağırımı öğrenme oranı = 0,0004 ise ikinci çağırıda homojenlikten ötürü öğrenme oranı = 0,0007 gelmektedir. Yapılan çalışmada kullanılan TPE algoritmasından dolayı değerler exploration sonrası belirli bir eşik değere yakın oluşturulmaya başlanmaktadır.

**hp.loguniform(etiket, alt, üst):** Bu fonksiyon etiket, alt ve üst olmak üzere üç girdiden yararlanmaktadır. Fonksiyon, alt ve üst değerlerden yararlanarak kuvvet(uniform(alt,üst))'a göre oluşturulan bir değer döndürmektedir. Dönüş değerinin logaritmasının düzgün dağılmasını sağlamaktadır. Optimizasyon işlemleri sırasında tanım aralığının belirlenmesi için kullanılan fonksiyona ait formül Eşitlik 3.1 ile gösterilmiştir.

$$\text{tanımaralığı} = [\exp(\text{alt}), \exp(\text{üst})] \quad (3.1)$$

Örnek: `hp.loguniform`(öğrenme katsayısı çarpanı, -0,5, 0,5) için Öğrenme katsayısı çarpanı tanım aralığı = [0,6065, 1,6498]. Fonksiyon bu aralıktan yararlanarak homojen olacak şekilde rastgele değer döndürmektedir. Yapılan çalışmada kullanılan TPE algoritmasından dolayı değerler exploration sonrası belirli bir eşik değere yakın oluşturulmaya başlanmaktadır.

**`hp.quniform`(etiket, düşük, yüksek, q):** Bu fonksiyon etiket, düşük, yüksek ve q olmak üzere dört girdiden yararlanmaktadır. Hem yukarı hemde aşağı sınırlandırılması gereken ayrık değerlerde yumuşatma (smoothing) yapmak için kullanıma uygundur. Fonksiyona ait formül Eşitlik 3.2 ile gösterilmiştir. Yuvarla fonksiyonu en yakın integer değeri döndürmektedir.

$$\text{parametre değeri} = \text{round} \left( \frac{\text{uniform}(\text{düşük}, \text{yüksek})}{q} \right) * q \quad (3.2)$$

Örnek: `hp.quniform`(artık\_birimler, 1-0.499, 4+0,499, 1) için

[0,511 , 4,499] aralığından rastgele değer döndürülür. Örnek olarak: 0,678. Sonrasında formülde yerine konulur.

$$\text{Yuvarla}(0,678/1) = 1$$

$$1 \times 1 = 1$$

`artık_birimler` tanım aralığı = [1,4]. Fonksiyon bu aralıktan homojen bir şekilde yararlanarak rastgele integer değer elde etmektedir. Yapılan çalışmada kullanılan TPE algoritmasından dolayı değerler exploration sonrası belirli bir eşik değere yakın oluşturulmaya başlanmaktadır.

### 3.4. Hiperparametre Uzayı

Bu kısımda tanımlanan hiperparametre uzayında bulunan her bir hiperparametre için parametrenin amacı, parametrik ifadesi, tanım aralığı hakkında bilgi verilmiştir.

**lr\_rate\_mult:** Bu parametre öğrenme oranının artırılması yada azaltılması için kullanılmıştır. Parametre, varsayılan öğrenme oranı (0,001) ile çarpıldıktan sonra o anki üzerinde çalışılan modele dinamik bir şekilde öğrenme oranı oluşturulması amacıyla kullanılır.

Tanımı : `hp.loguniform(lr_rate_mult,-0,5, 0,5)`

Tanım aralığı = [0,6065, 1,6498]

**l2\_weight\_reg\_mult:** Bu parametre öğrenme l2 ağırlıklarının düzenlenmesi için kullanılmıştır. Genel amacı DL sırasında aşırı ezberleme ihtimalini azaltmaktadır. Parametre varsayılan değer (0,0007) ile çarpılarak çeşitli katmanların ağırlıklarını düzenlemek için kullanılmaktadır. Parametre, keras yardımıyla evrişimli katmanlar, yoğun (Dense) katmanlar, tam bağlı katmanlar, ayrık (Residual) katmanlar ve evrişimli havuzlama yöntemi üzerinde kullanılmıştır.

Tanımı : `hp.loguniform('l2_weight_reg_mult', -1,3, 1,3)`

Tanım aralığı = [0,272532, 3,669297]

**batch\_size:** Bu parametre toplu iş örnek sayısı (batch size) belirlenmesi için kullanılmaktadır. Modellerin eğitim aşaması öncesinde parametre uzayından çekilir ve eğitim sırasında kullanılır.

Tanımı : `hp.quniform('batch_size', 10, 32, 2)`

Tanım aralığı = [10,32]

**optimizer:** Parametre, öğrenme oranı değerinin değiştirmek için hangi algoritmanın kullanılması gerektiğine karar vermek için kullanılmaktadır.

Tanımı : `hp.choice('optimizer', ['Adam', 'Nadam', 'RMSprop'])`

Tanım aralığı = {'Adam', 'Nadam', 'RMSprop'}

**conv\_dropout\_drop\_proba:** Parametre, evrişim katmanlarını sonrası kullanılma ihtimali bulunan sönümlenme katmanına ait sönümlenme olasılığının belirlenmesi için kullanılmıştır.

Tanımı : `hp.uniform('conv_dropout_proba', 0,0, 0,35)`

Tanım aralığı = [0, 0,35]

Bu parametre sadece evrişim katmanları sonrası aktive olmaktadır. Hiperparametre uzayından 0.0 değeri döndürülmesi durumunda evrişim sonrası sönümlenme katmanı kullanılmamaktadır.

**fc\_dropout\_drop\_proba:** Parametre, tam bağlı katmanların sonrası kullanılma ihtimali bulunan sönümlenme katmanına ait sönümlenme olasılığının belirlenmesi için kullanılmıştır.

Tanımı : `hp.uniform('fc_dropout_proba', 0,0, 0,6)`

Tanım aralığı = [0, 0,6]

Bu parametre sadece tam bağlı katmanlar sonrası aktive olmaktadır. Hiperparametre uzayından 0.0 değeri döndürülmesi durumunda tam bağlı katman sonrası sönümlenme katmanı kullanılmamaktadır.

**use\_BN:** Parametre, toplu normalleştirme katmanı kullanılma ihtimalinin belirlenmesi için kullanılmıştır.

Tanımı : `hp.choice('use_BN', [False, True])`

Tanım aralığı = 'True', 'False

Bu parametreden evrişimli katmanlar, ayırık katmanlar, evrişimli havuzlama sonrasında yararlanılabilir

**first\_conv:** Parametre, modelin başlangıcında özel bir evrişimli katman yapısı kullanılmasına yada kullanılmamasına karar vermektedir. Eğer kullanılacaksa evrişim katmanındaki kernel büyüklüğü bu parametre yardımıyla belirlenmektedir.

Tanımı : `hp.choice('first_conv', [None, hp.choice('first_conv_size', [3, 4])])`

Tanım aralığı = 'True', 'False

Modelin ilk oluşturma aşamasında parametre uzayından 'None' gelmesi halinde bu parametre üzerinde işlem yapmadan atlanır. Diğer durumlarda dönen değer sayısı kadar evrişimli katmanı arka arkaya olacak şekilde modelin başlangıcına yerleştirilmesini sağlamaktadır.

**nb\_conv\_pool\_layers:** Parametre, arka arkaya kullanılacak evrişimli, havuzlama katmanı sayısını belirlemektedir.

Tanımı : `hp.choice('nb_conv_pool_layers', [2, 3])`

Tanım aralığı = [2,3]

first\_conv parametresi sonrası, kaç kez evrişim ve havuzlama katmanlarından oluşan yapının kullanılacağına karar vermektedir.

**conv\_pool\_res\_start\_idx:** Parametre, artık ağ bağlantısı kullanılmasına başlamak için gerekli olan model derinliğini belirlemek için kullanılmıştır.

Tanımı : `hp.quniform('conv_pool_res_start_idx', 0, 2, 1)`

Tanım aralığı = [0,2]

Bu parametrenin, aktive olabilmesi için `nb_conv_pool_layers` parametresinin değerinin kendisinden daha büyük olması gerekmektedir. Uygun derinlik aşıldıktan sonra `residual` parametresinin aktif olup olmamasını denetlemektedir.

**conv\_hiddn\_units\_mult:** Parametre, evrişim ve artık ağ katmanlarında kullanılacak filtre sayısının belirlenmesi için kullanılmaktadır. Varsayılan değer (16) ile bu parametreden gelen değer çarpımı sonucu filtre sayısı belirlenmektedir.

Tanımı : `hp.loguniform('conv_hiddn_units_mult', -0.5, 0.5)`

Tanım aralığı = [0,6065, 1,6498]

**residual:** Parametre, artık ağ bağlantısı kullanılmasına yada kullanılmamasına karar vermektedir. Eğer kullanılacaksa kaç tane bağlantı arka arkaya kullanılacak bunu belirlemektedir.

Tanımı : `hp.choice('residual', [None, hp.quniform('residual_units', 1 - 0.499, 4 + 0.499, 1)])`

Tanım aralığı = {None, '1', '2', '3', '4'}

Bu parametre `conv_pool_res_start_idx` parametresi tarafından belirlenen derinlik geçildikten sonra kullanılabilir hale gelmektedir. Modelin ilk oluşturma aşamasında parametre uzayından 'None' gelmesi halinde bu parametre üzerinde işlem yapmadan atlanır. Diğer durumlarda dönen değer sayısı kadar artık ağ bağlantısını arka arkaya olacak şekilde oluşturulmasını sağlamaktadır.

**pooling\_type:** Parametre, hangi havuzlama katmanı (maksimum havuzlama, ortalama havuzlama, tamamen evrişimsel havuzlama, inception havuzlama) tipinin kullanılacağına karar vermektedir. Tez kapsamında maksimum havuzlama, ortalama havuzlama, tamamen-evrişimsel havuzlama (Springenberg vd., 2014), inception havuzlama (Szegedy vd., 2017) yöntemlerinden yararlanılmıştır.

Tanımı : `hp.choice('pooling_type', ['max', 'avg', 'all_conv', 'inception'])`,

Tanım aralığı = {'max', 'avg', 'all\_conv', 'inception'}

Bu parametre, parametre uzayı tarafından her model için bir kez belirlenmektedir. Model üretimi sırasında kullanılacak tüm havuzlama katmanlarında aynı tür kullanılmaktadır.

**conv\_kernel\_size:** Parametre, evrişimli katmanlarda kullanılacak çekirdek büyüklüğünü belirlemektedir.

Tanımı : `hp.quniform('conv_kernel_size', 2, 4, 1)`

Tanım aralığı = [2,4]

**res\_conv\_kernel\_size:** Parametre, artık ağ katmanlarında kullanılacak çekirdek büyüklüğünü belirlemektedir.

Tanımı : `hp.quniform('res_conv_kernel_size', 2, 4, 1)`

Tanım aralığı = [2,4]

**fc\_units\_1\_mult:** Bu parametre tam bağlı katmanlarda bulunan nöronların artırılması yada azaltılması için kullanılmıştır. Parametre, varsayılan tam bağlı katman nöron sayısı (300) ile çarpıldıktan sonra o anki üzerinde çalışılan modele dinamik bir şekilde tam bağlı katman nöron sayısı oluşturulması amacıyla kullanılmaktadır.



Tanımı : `hp.loguniform('fc_units_1_mult', -0,6, 0,6)`

Tanım aralığı = [0,5488, 1,8221]

**one\_more\_fc:** Bu parametre tam bağı katman sonrasında başka bir tam bağı katman kullanılmasına yada kullanılmamasına karar vermektedir. Eğer kullanılacaksa yeni tam bağı katmanın nöron sayısı varsayılan değer (200) bu parametre ile çarpıldıktan sonra oluşturulmaktadır.

Tanımı : `hp.choice('one_more_fc', [None, hp.loguniform('fc_units_2_mult', -0.6, 0.6)])`

Tanım aralığı = {'None'} U [0,5488, 1,8221]

**activation:** Bu parametre modelin üretimi sırasında hangi aktivasyon fonksiyonunun kullanılacağını belirlemektedir. Parametreden evrimsel katmanlar, artık ağ katmanları ve tam bağı katmanlarda yararlanılmaktadır.

Tanımı : `hp.choice('activation', ['relu', 'gelu', 'swish'])`

Tanım aralığı = 'relu', 'gelu', 'swish'

### 3.5. Ağaç Bazlı Parzen Tahminleyicileri

Hiperparametre optimizasyonu için ağaç bazlı parzen tahminleyicileri (Bergstra vd., 2011) algoritmasından yararlanılmıştır. Bu yöntem Bayes optimizasyonu yöntemidir ve çeşitli Kaggle yarışmalarının kazanılmasında hiperparametre optimizasyonu gerçekleştirilmesi için kullanılmıştır (Watanabe, 2023). Algoritma şu adımlardan oluşmaktadır :

- Başlangıç değerlerinin ayarlanması
- Rastgele değer için amaç fonksiyonunun değerlendirilmesi
- Ayırma algoritmasının kullanımı
- Ağırlıklandırma algoritmasının kullanımı

- Uygun bant genişliğinin seçimi
- Parametre tipine uygun bilgilendirici olmayan önceki (Non-Informative Prior) belirlenerek, sömürü (exploitation) kademesinin çok yüksek olmasının önlenmesi
- Edinme fonksiyonu tarafından değerlendirme yapılarak yeni değerlerin seçilmesi
- Yeni değer kullanılarak amaç fonksiyonunun değerlendirilmesi

TPE ile sırayla oluşturulan modellerin performanslarına bakılarak geçmişte modele karşılık gelen hiperparametrelerden hangilerinin daha verimli sonuçlar elde ettiği hesaplanarak bir eşik değeri seçilmekte sonraki çalışma zamanında bu eşik değere yakın değerler seçilmesi amaçlanmaktadır. Eşik değerinin seçimi, başlangıçta keşif (exploration) düşüncesi ile değiştirilmesi kolay, yumuşak sınırlar çerçevesinde yapılmakta sonrasında uygun değer aralığı bulunduğu sömürü (exploitation) aşamasına geçilerek değiştirilmesi zor, katı bir sınır çerçevesi içerisinde gerçekleştirilmektedir.

### **3.6. Başlangıç Parametreleri**

Bu kısımda modellerin eğitilmesi için kullanılan başlangıç parametrelerine ve bu parametreleri etkileyen bir hiperparametre varsa bu parametrenin hangisi olduğuna dair bilgilere Tablo 1’de yer verilmiştir. Hiperparametre optimizasyonu sayesinde başlangıç değerleri optimize edilerek, başlangıç değerlerinin model mimarisine uygun olmadığı durumlarda bile verimli doğruluk skoru elde edebilen modellerin oluşturulması sağlanmıştır. Başlangıç değerleri olarak literatürde sıklıkla yararlanılan değerler kullanılmıştır.

Tablo 1

Modellerin üretimi aşamasında kullanılan başlangıç parametreleri ve bu parametreleri etkileyen hiperparametreler tablosu

Parametre	Hiperparametre	Değer
Öğrenme oranı	lr_rate_mult	'0,001'
Adım sayısı	-	'100'
l2 öğrenme oranı	l2_weight_reg_mult	'0,0007'
Filtre sayısı	conv_hiddn_units_mult	'16'
Tam bağlı katman nöron sayısı	fc_units_l_mult	'300'
İkinci tam bağlı katman nöron sayısı	one_more_fc	'200'

### 3.7. Ön İşleme Aşaması ve Eğitim Prosedürü

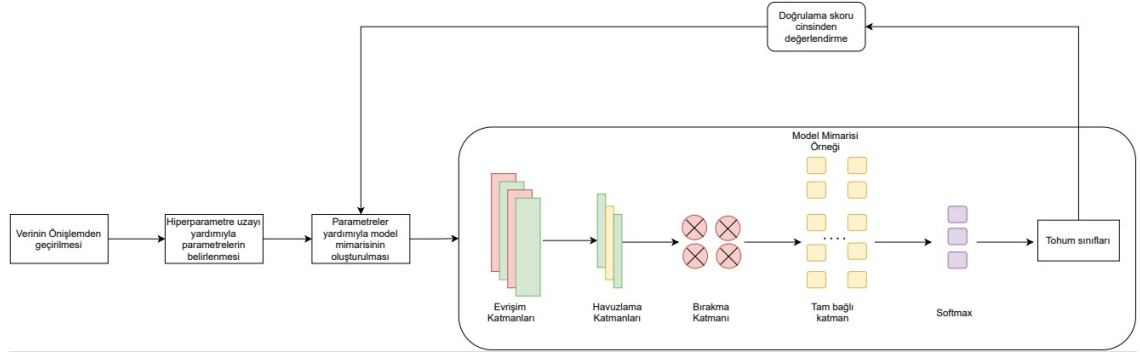
Bu kısımda veri kümesinin modellerin eğitimi için kullanılabilmesi için gerçekleştirilen ön işlemler anlatılmıştır. Veri kümesinde bulunan tohum görüntülerinin piksel sayıları tohum morfolojisine bağlı olarak değişmektedir. Oluşturulan python kodundan yararlanarak tüm görüntüler 224x224 piksel olacak şekilde getirilmiştir. Hedef olarak 224x224 seçilmesinin sebebi 2019'da (Altuntaş vd., 2019) yapılan çalışmada en iyi sonucu üreten modelin VGG-19 olması ve bu modelin 224x224'lük girdi kullanmasından kaynaklanmaktadır. Bir sonraki aşamada 3000 tohum görüntüsü 2400'ü eğitim 600'ü test için olacak şekilde rastgele iki gruba ayrılmıştır. Eğitim veri kümesi içerisinde 1416 diploid 984 haploid, test veri kümesi içerisinde 354 diploid 246 haploid tohum görüntüsü içermektedir. Görüntüler, python üzerinde okunmuş, numpy dizisine çevrilmiş ve görüntülerdeki üç kanal (RGB) üzerinde normalizasyon işlemi gerçekleştirilmiştir. Normalizasyon işlemi için görüntülere ait pikseller 255'e bölünmüş oluşan değerlerden 0.5 çıkartılarak görüntüye ait piksel değerlerinin 0 etrafına kümelenebilmesi sağlanmıştır. Eğitim öncesi veriler lambda adı verilen bir katman aracılığıyla tensorflowa ait random\_flip\_left\_right fonksiyonu yardımıyla veri artırımı gerçekleştirilmiştir. Bu fonksiyon ile görüntülerin rastgele olacak şekilde yatay düzlemde soldan sağa döndürülmesi sağlanmıştır. Sonraki aşamada önceki kısımda verilen başlangıç parametreleri kullanılarak

modellerin oluřturulma, eđitilme ve dođrulama skoru elde edilmesi iřlemi bařlatılır. Kullanılan kayıp fonksiyonu her model iin aynı olup Kategorik apraz Entropi (categorical crossentropy)'dir. Modellerin retimi ařamasında keras ktphanesinin zelliđi olan callback'ten yararlanılarak her model iin her adıma ait dođrulama skoru kayıt edilmiř ve eđitim, test iřlemi sonucu modele ait en iyi dođrulama skoru gz nnde bulundurarak modellerin verimlilikleri tartiřılmıřtır. rnek olarak veri kmesi zerinde %94,66 dođrulama skoru elde eden model, 100 adım eđitilmesine rađmen 70. adımda bu sonuca ulařmıřtır.

Modellerin tamamı iin sadece evriřimsel ve inception havuzlama, katmanlarında kullanılan stride deđeri '2' olarak atanmıřtır. Geri kalan katmanların (Evriřimsel, artık) tamamında stride deđeri '1' dir. Modellerin tamamı iin katmanlara ait padding deđeri 'same' dir.

### **3.8. Modellerin retim Adımları**

Bu kısımda modellerin retimi sırasında gerekleřen iřlemler anlatılmıř ve mimarilerin her adımda nasıl optimize edildiđinden bahsedilmiřtir. ncelikli olarak veriler niřlemden geirilerek veri kmesi zerinde veri artırımını (Data Augmentation) yapılmıřtır. Sonrasında daha nceden tanımladıđımız hiperparametre uzayından yararlanarak TPE algoritması tarafından parametre grubu iřlemi gereklenmektedir. Bu parametre grubu yardımıyla model mimarisi retilmekte ve mısır tohumlarına ait veri kmesi kullanılarak ikili sınıflandırma iřlemi gereklenmektedir. Sonraki adımda sınıflandırma iřlemine ait dođrulama skoru deđerlendirilerek, TPE yardımıyla yeni bir parametre grubu oluřturulmakta ve yeniden aynı iřlemler tekrarlanmaktadır. Őekil 7 modellerin retim adımlarını gstermektedir.



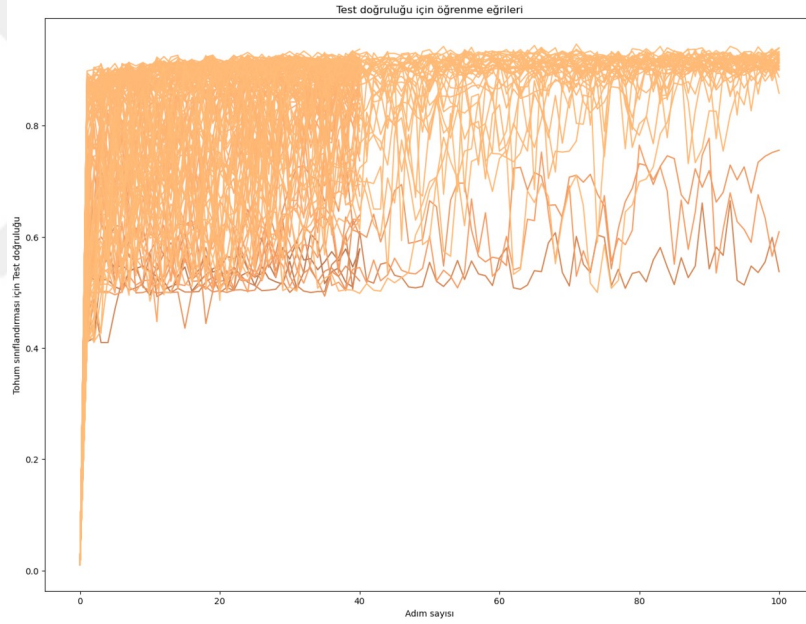
Şekil 7. Modellerin üretim adımları.

Şekil 7 üzerinde görülebilecek model mimarisi örneğinde bulunan sönümlenme (bırakma) katmanı hiperparametre uzayından gelecek değerlerin tamamı için oluşmamaktadır. TPE algoritmasından dönen parametre değerlerine bağlı olarak evrişim katmanı sonrası ve tam bağlı katman sonrası sönümlenme katmanı kullanılma olasılığı bulunmaktadır. Mimarinin üretimi sırasında kullanılan evrişim katmanı, havuzlama katmanı ve tam bağlı katman sayısında daha önceden tanımlanan hiperparametre uzayından TPE algoritması yardımıyla döndürülen değerler yardımıyla oluşturulmaktadır.

## DÖRDÜNCÜ BÖLÜM

### ARAŞTIRMA BULGULARI

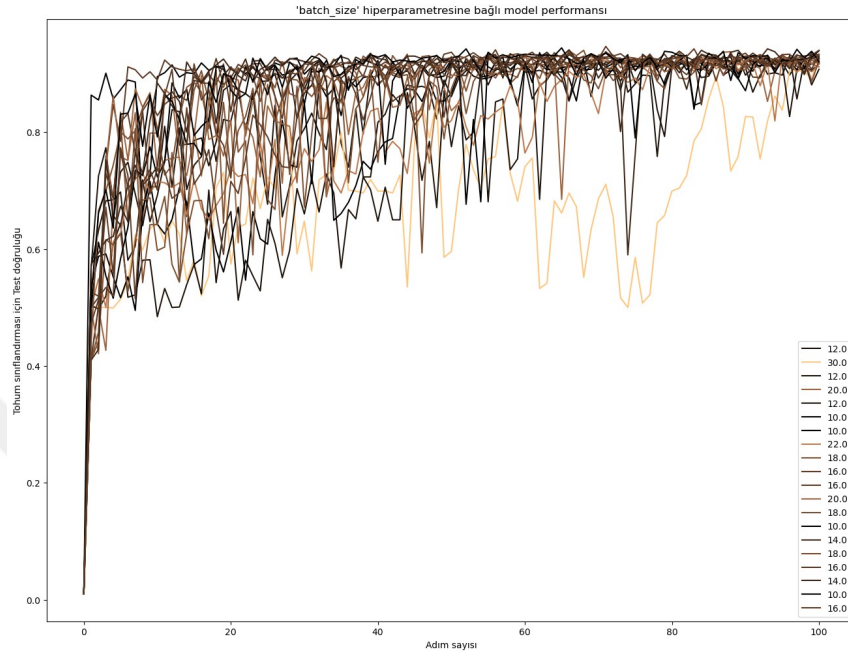
Tez çalışmasında mısır tohumlarının sınıflandırılması için melez parametre optimizasyonundan yararlanılarak ESA bazlı model mimarisi üretimi işlemi gerçekleştirilmiştir. Bu bölümde tanımladığımız hiperparametre uzayı kullanılarak Hyperopt kütüphanesinin yardımıyla TPE algoritması ile oluşturulan modeller test doğruluğu açısından karşılaştırılmıştır. Üretilen 100 modelin en iyi 20'si üzerinden hangi parametrelerin, test doğruluğuna olumlu yönde etki ettiği incelenmiş ve parametrelerin etkisi tartışılmıştır. Son olarak üretilen modeller arasından test doğruluğu açısından en yüksek skoru elde eden modelin mimarisi paylaşılmıştır.



Şekil 8. Üretilen tüm modeller için adım sayısına karşılık gelen test doğruluğu için öğrenme eğrileri.

Şekil 8'de üretilen 150 modelin her birine ait test doğruluğu grafiği yer almaktadır. Elde edilen en düşük skor 0,5916, en yüksek skoru 0,9466 olmuştur. Problemin ikili sınıflandırma işlemi olduğu göz önünde bulundurulursa en kötü sonuç elde eden modelin verimsiz çalıştığı görülebilir. Grafiklerin okunabilirliğini arttırmak ve görece olarak verimsiz çalışan modellere ait verileri dikkate almamak için bu aşamadan sonra test

doğruluğu açısından en iyi 20 skoru üreten model ve bu modellerin parametreleri hakkında bilgi verilecektir. En iyi 20 model arasından en kötü sonuç eden modelin test doğruluğu skoru 0,9308'dir. Kalan modellerin skorları 0,9308 – 0,9466 aralığında yer almaktadır.



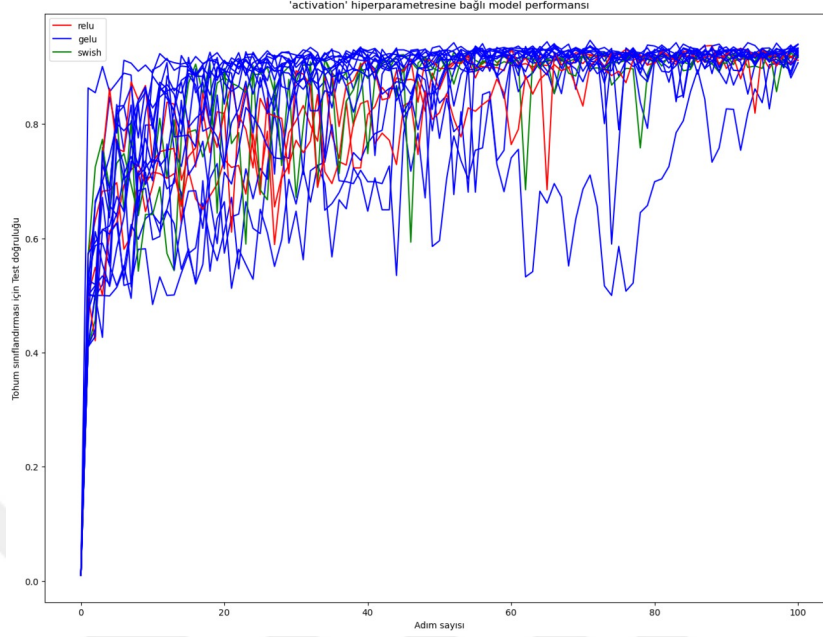
Şekil 9. En iyi 20 model için batch\_size hiperparametresi sonuçları.

Şekil 9'da batch\_size parametresi için model sonuçları yer almaktadır. Grafik incelendiğinde 10-32 aralığında değer alabilen parametrenin çoğunlukla üst sınıra yakın değerlere sahip olmadığı görülmüştür. Bu durumdan ötürü batch\_size değerinin düşük olmasının modellerin verimliliği açısından pozitif etkisi olduğu söylenebilir.

Şekil 10'da activation parametresi için model sonuçları yer almaktadır. En iyi 20 modelden büyük çoğunluğu gelu aktivasyon fonksiyonundan yararlanmışır. Sırasıyla bunu relu ve swish takip etmiştir. Grafığe bakılarak oluşturduğumuz hiperparametre uzayı için GELU aktivasyon fonksiyonunun daha verimli sonuçlar elde ettiği söylenebilmektedir.

GELU'nun neden ReLU'dan daha iyi performans gösterdiğine dair olası bir açıklama, GELU'nun sifıra yakın yerlerde daha pürüzsüz olması ve tüm aralıklarda türevlenebilir olmasıdır. GELU, negatif aralıkta küçük gradyanlara izin verdiği için ReLU'nun yaşadığı

"ölen nöronlar" sorununun kötü yönlü etkisini hafifletmektedir (Hendrycks ve Gimpel, 2016).

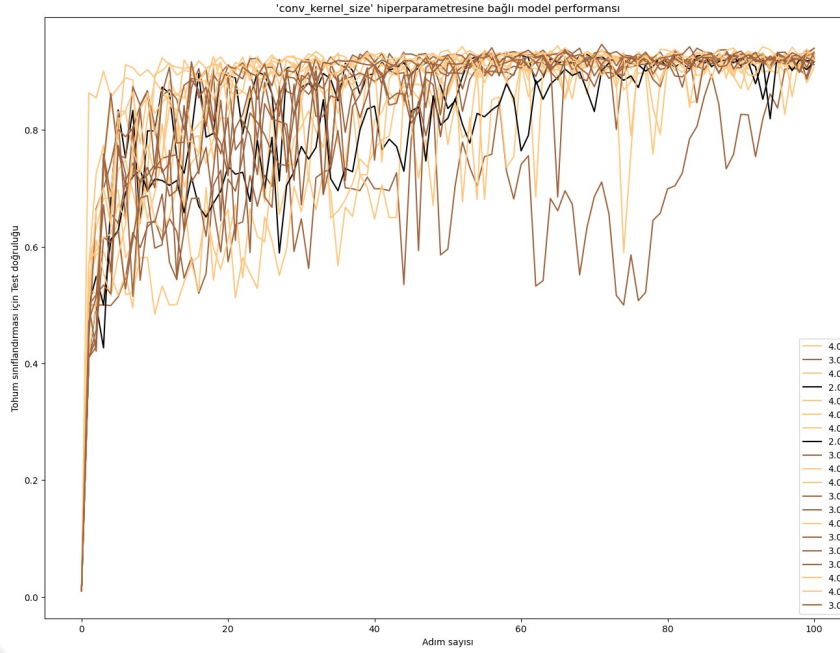


Şekil 10. En iyi 20 model için activation hiperparametresi sonuçları.

Çekirdek boyutunun artması toplam parametre sayısını arttırmaktadır. Bu durum sınıflandırmayı gerçekleştirmek için kullanılan modelin daha yüksek bir karmaşıklığa sahip olmasına yol açmaktadır. Ancak yüksek çekirdek boyutu her zaman daha iyi verimlilik sağlamamaktadır. Daha önceden yapılan çalışmalar (Richter vd., 2021) en iyi verimli sonucu elde etmek için çekirdek boyutunun her ağ mimarisine özel olarak oluşturulması gerektiğini bulmuştur.

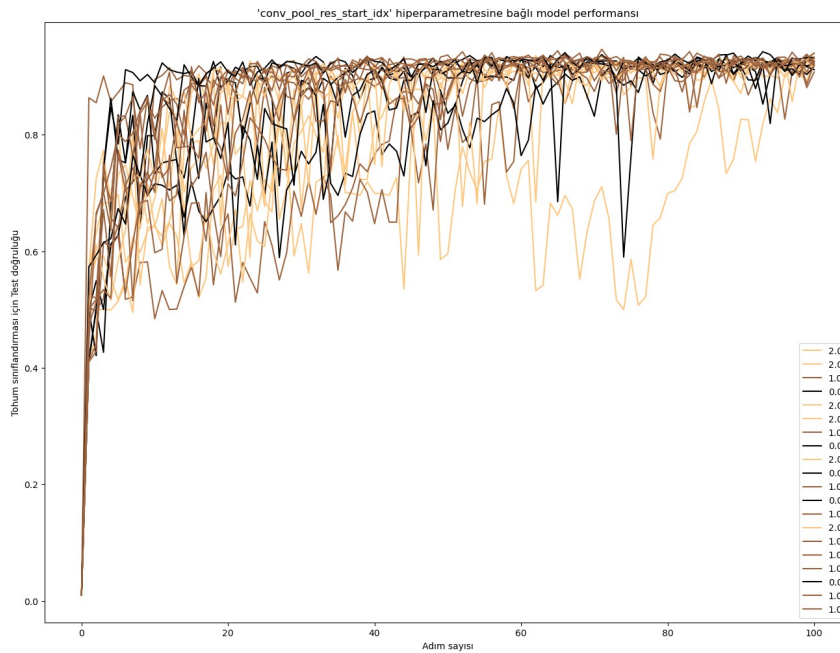
Şekil 11’de conv\_kernel\_size parametresi için model sonuçları yer almaktadır. Grafik incelendiğinde çekirdek boyutu olarak 3 ve 4’ün belirgin bir şekilde daha fazla kullanıldığı görülmüştür. Grafikte evrişimli katmanlar için çekirdek büyüklüğünün ikiden büyük olmasının test doğruluğu yönünden pozitif etkisi görülmüştür. Yeni oluşturulacak hiperparametre uzayında 3 ve 4 ‘ün yanı sıra daha yüksek çekirdek boyutları kullanılması önerilmektedir.





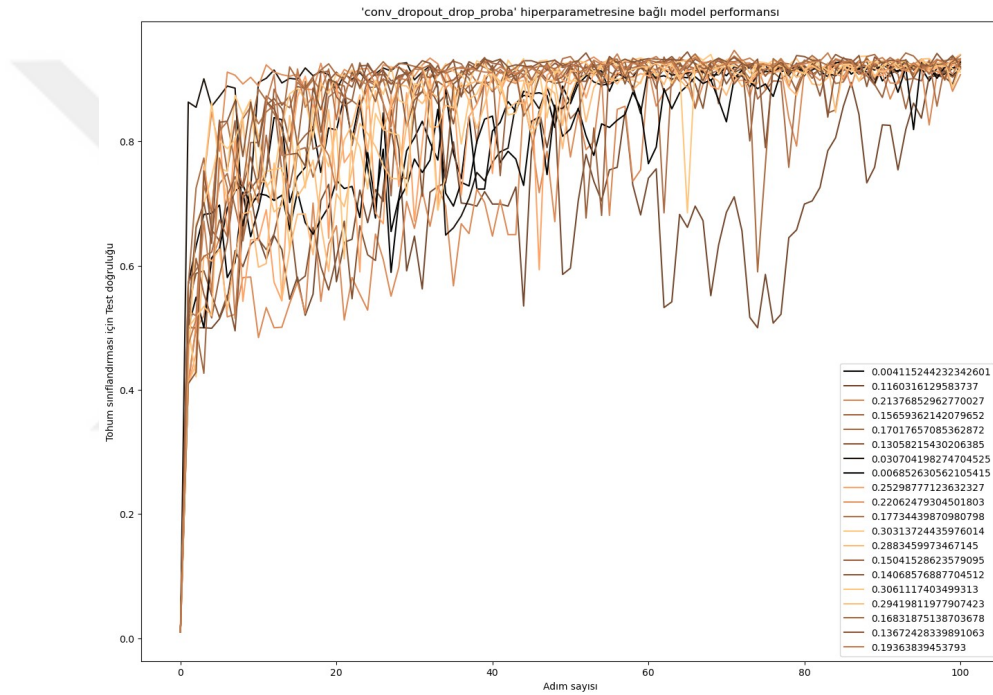
Şekil 11. En iyi 20 model için `conv_kernel_size` hiperparametresi sonuçları.

Şekil 12’de `conv_pool_res_start_idx` parametresi için model sonuçları yer almaktadır. Bu parametre için grafikten anlamlı bir sonuç bulunamamıştır. Tanım aralığında bulunan her bir deđer [0,1,2] için yaklaşık benzer sayıda (5-8) kullanım vardır



Şekil 12. En iyi 20 model için `conv_pool_res_start_idx` hiperparametresi sonuçları.

Şekil 13'te `conv_dropout_drop_proba` parametresi için model sonuçları yer almaktadır. Bu grafikte evrişimli katmanlardan sonra sönümlenme katmanı kullanılırsa bu katmandaki sönümlenme oranı kaç olmalı ? Sorusuna cevap aranmıştır. Üst limit olan 0,35 göz önüne alındığında verimli çalışan modellerin bu değere büyük ölçüde yakınlaşmadığı görülmüştür. Modellerin 13 tanesinde 0,20 altı değerler kullanılmıştır. Tanımladığımız hiperparametre uzayının bu açıdan istenenden büyük bir çerçeveye sahip olduğu söylenebilir. Yeni bir hiperparametre uzayı oluşturulması sırasında bu parametre için [0, 0,20] aralığında tanım yapılması önerilmektedir.

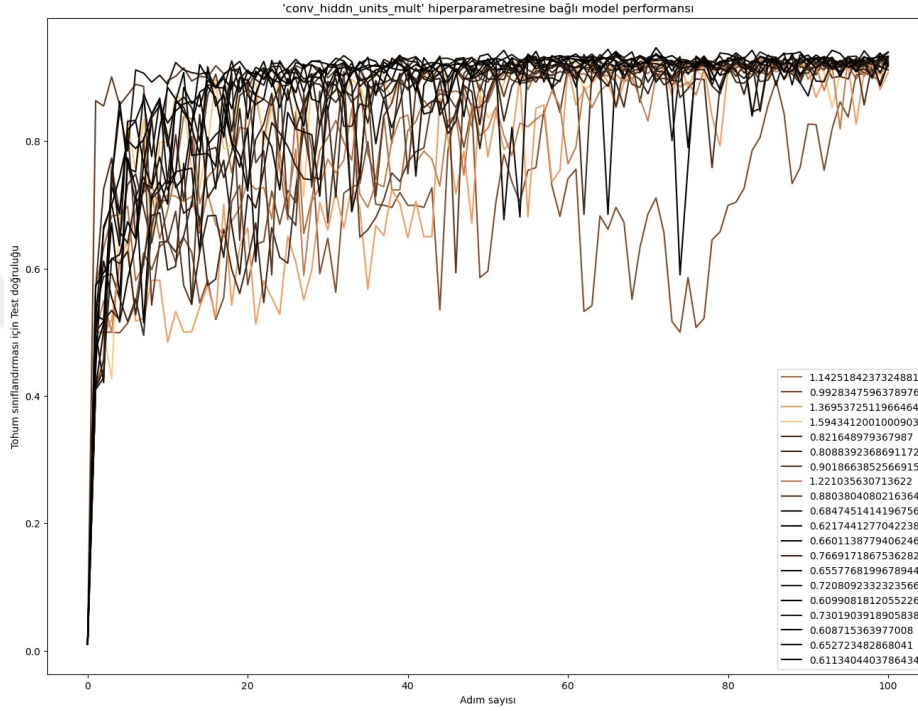


Şekil 13. En iyi 20 model için `conv_dropout_drop_proba` hiperparametresi sonuçları.

Evrişimsel katmandaki optimum filtre sayısı sabit bir değer değildir ve çözülmesi istenen probleme ve veri kümesine göre değişebilmektedir. Filtre sayısının çokluğu ağıın öğrenebileceği özellik miktarını arttırmaktadır.

Şekil 14'te `conv_hidden_units_mult` parametresi için model sonuçları yer almaktadır. Parametre evrişim katmanlarındaki filtre sayısının belirlenmesi için kullanılmaktadır. Tanım aralığı [0,6065, 1,6498] ve varsayılan değer olan 16 göz önüne

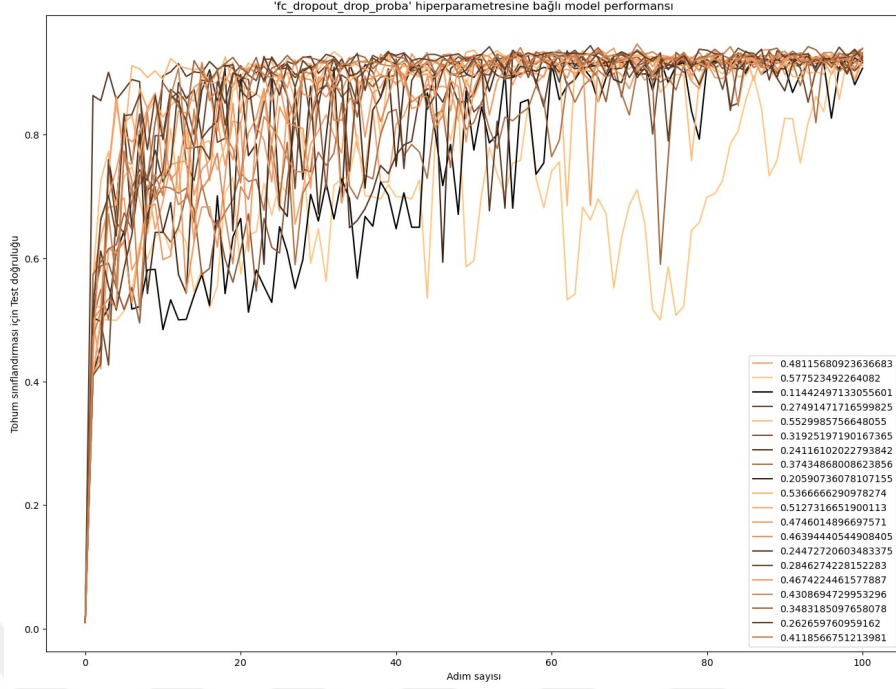
alındığında. Grafikten görülebileceği üzere en iyi modellerin çoğunluğu alt sınıra yakın değerlerden yararlanmıştır. Bu parametre için varsayılan değerın düşürülmesi ya tanım aralığının orta noktası 0,8 olacak şekilde yeniden belirlenmesi önerilmektedir. Elde edilen sonuçlar ışığında mimarilerin oluşturulması sırasında kullanılan filtre sayısının düşürülebileceği görülmüştür.



Şekil 14. En iyi 20 model için conv\_hidden\_units\_mult hiperparametresi sonuçları.

Sönümlenme oranının çok yüksek olması yetersiz uyumlamaya (underfitting) yol açabilmektedir. Bu oranın çok az olması ise aşırı uyumu (overfitting) önlemek için yeterli düzenleme sağlanmasını engellemektedir (Park ve Kwak, 2017).

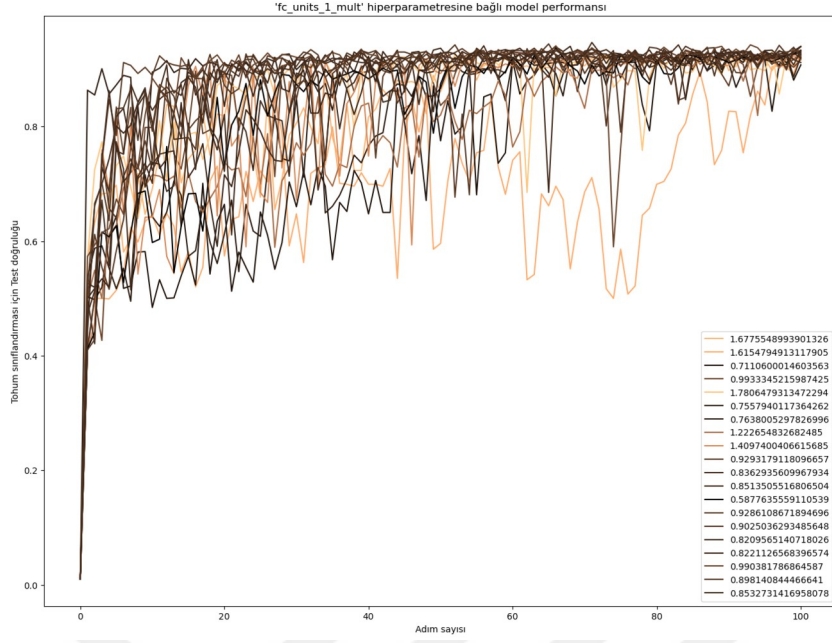
Şekil 15'te fc\_dropout\_drop\_proba parametresi için model sonuçları yer almaktadır. Grafik incelendiğinden 10 farklı model için parametrenin 0,40 üzerinde olduğu ve sadece bir modelde parametrenin 0,20'nin altında olduğu görülmüştür. Sonuçlara bakarak tanım aralığının [0, 0,6] olduğu göz önünde bulundurulursa tanım aralığının [0,20, 0,80] olacak şekilde yeniden düzenlenmesi gerektiği düşünülmektedir.



Şekil 15. En iyi 20 model için `fc_dropout_drop_proba` hiperparametresi sonuçları.

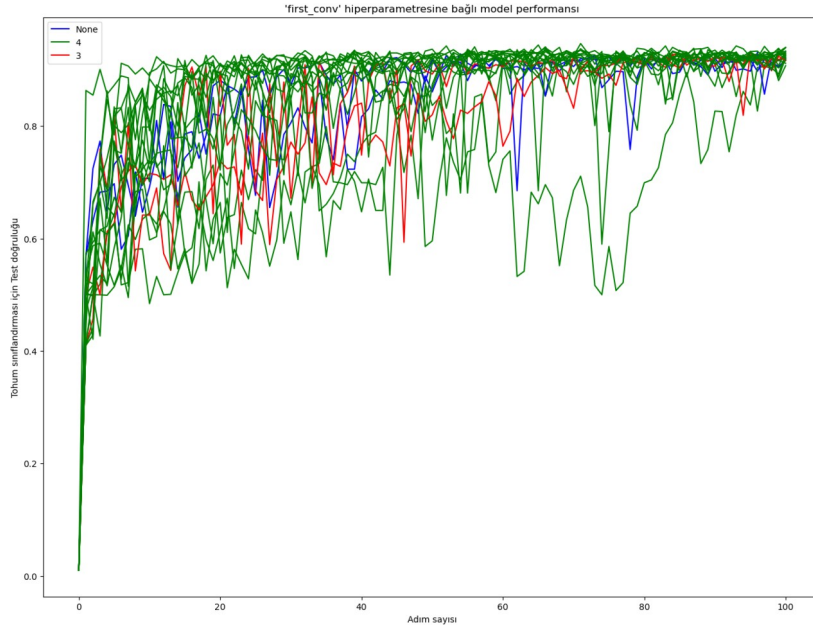
Tam bağlantılı katmanın amacı, belirli bir görev için evrişimli katmanlar tarafından tespit edilen özellikleri birleştirmektir. Katmanlarda bulunan nöron sayısı modelin performansı büyük ölçüde etkilemektedir. Nöron sayının artması girdi verileri ile hedef değişkenler arası ilişkiler yumağının anlaşılmasını kolaylaştırmaktadır. Ancak sayının fazla olması durumunda model karmaşıklığı yüzünden aşırı ezberlemeye yol açabilmektedir.

Şekil 16'da `fc_units_1_mult` parametresi için model sonuçları yer almaktadır. Bu parametre tam bağlı katmandaki nöron sayısını belirlemektedir. Parametrenin tanım aralığı olan  $[0,5488, 1,8221]$  göz önünde bulundurulduğunda grafikten 15 modelin 1 değerinin altında verimli sonuç ürettiği görülmüştür. Bundan dolayı ya parametre uzayı 1'e yaklaşacak şekilde yeniden tanımlanması yada varsayılan tam bağlı katman nöron sayısı 300'ün düşürülmesi önerilmektedir.



Şekil 16. En iyi 20 model için fc\_units\_1\_mult hiperparametresi sonuçları.

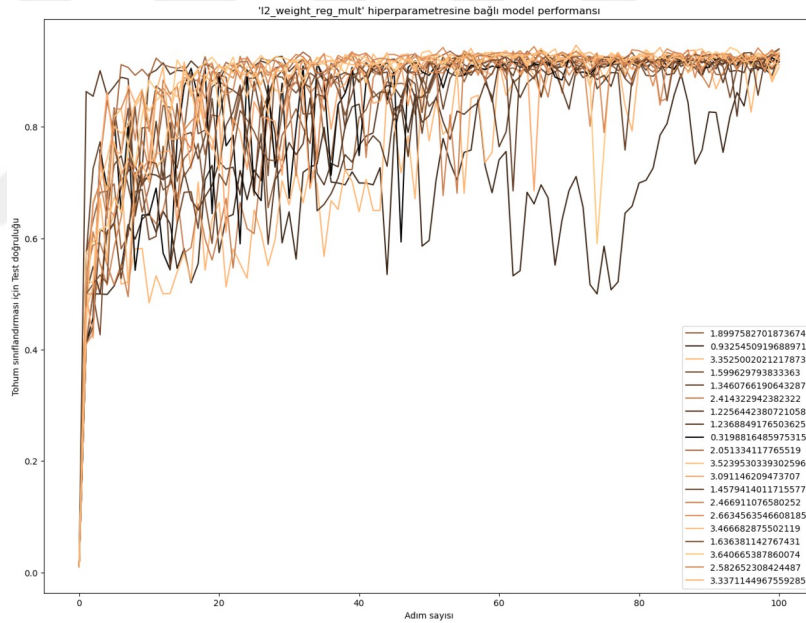
Şekil 17'de first\_conv parametresi için model sonuçları yer almaktadır. Bu parametre model başlangıcında özel evrişimli katman yapısı kullanılıp, kullanılmamasına karar vermektedir. Grafikten görülebileceği üzere 4 model dışında kalan tüm modellerde 4 parametresi kullanılmıştır. Yani en iyi sonuç üreten 16 modelde özel yapı olarak ilk 4 katman arka arkaya evrişim katmanından oluşmaktadır.



Şekil 17. En iyi 20 model için first\_conv hiperparametresi sonuçları.

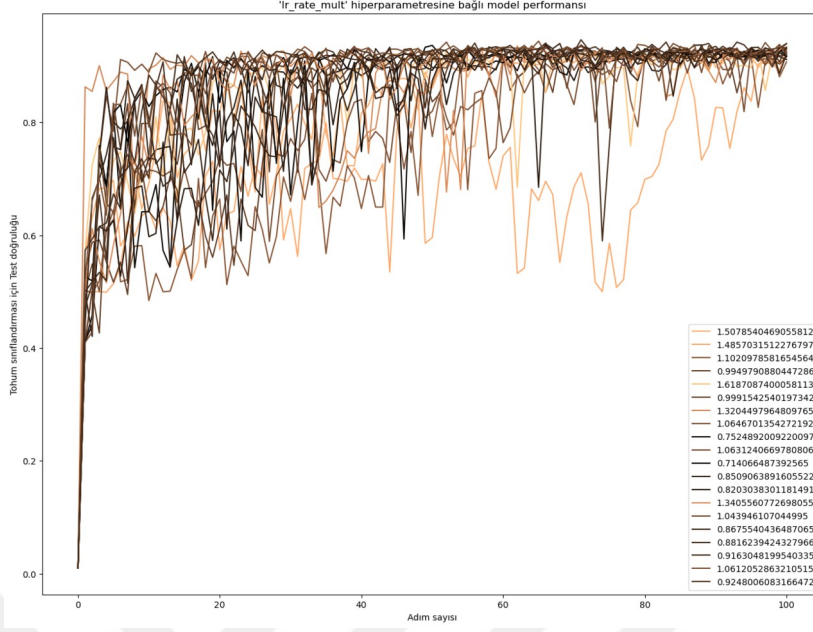
Sonuçlar ışığında first\_conv parametresinin tanım kümesinin {'none','3','4'} 'ten {'4','5','6'}'ya değiştirilmesi önerilmektedir. Bu değer için none parametresine sahip bir model dışında hiç bir model ilk 20'ye girememiştir. Sonuçlar incelendiğinde model mimarisinin başlangıcında kullanılan özel evrişim katmanı yapısının modellerin test doğruluk skorlarına büyük ölçüde olumlu etki ettiği görülmüştür.

Şekil 18'de l2\_weight\_reg\_mult parametresi için model sonuçları yer almaktadır. Tanım aralığı göz önüne alındığında [0,272532, 3,669297] 18 modelin parametre olarak 1'den yüksek değer aldığı görülmüştür. Bu parametrenin tanım aralığının yukarı yönlü olacak şekilde yeniden tanımlanması yada varsayılan değer olan 0.0007'nin artırılması önerilmektedir.



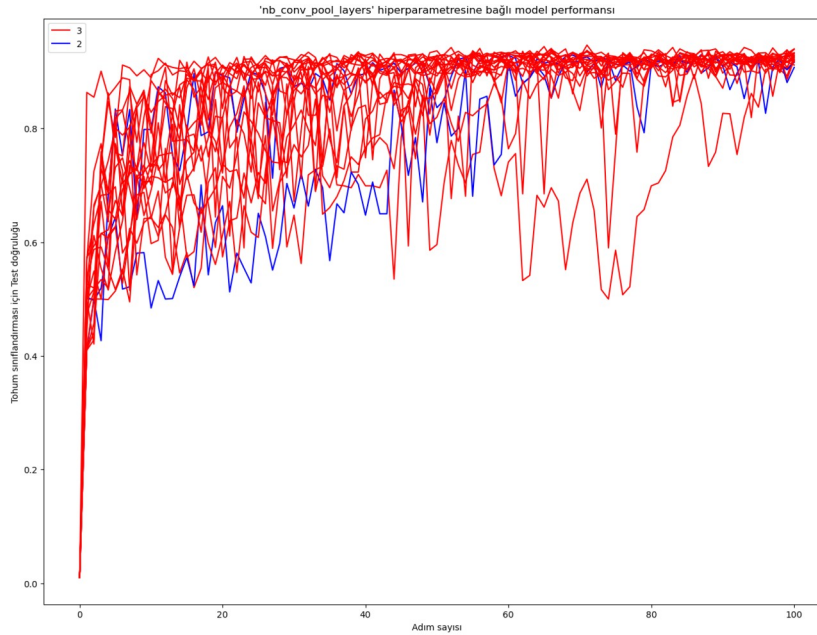
Şekil 18. En iyi 20 model için l2\_weight\_reg\_mult hiperparametresi sonuçları.

Şekil 19'da lr\_rate\_mult parametresi için model sonuçları yer almaktadır. [0,6065, 1,6498] tanım aralığı göz önüne alındığında bu parametreye ait sonuçların eşit dağıldığı söylenebilir. Grafikten bu parametreye dair belirgin bir örüntü yakalanamamıştır. Sonuçlar incelendiğinde öğrenme oranını uyarlanabilir bir şekilde belirleyen optimizasyon yöntemlerinin (ADAM, NADAM, RMSProp) verimli çalıştığı için TPE'e tarafından hesaplanan eşik değerinden büyük ölçüde etkilenmediği düşünülmektedir.



Şekil 19. En iyi 20 model için lr\_rate\_mult hiperparametresi sonuçları.

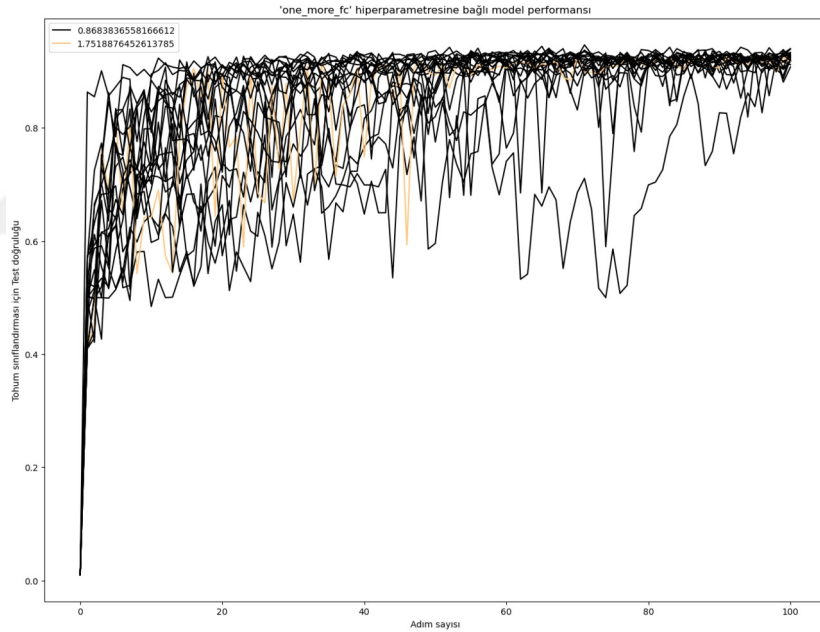
Şekil 20’de nb\_conv\_pool\_layers parametresi için model sonuçları yer almaktadır. Grafik incelendiğinde modeller 18 tanesinin bu parametre için 3 değerini kullandığı görülmüştür. {2,3} olan tanım aralığının {3,4,5,6} olarak genişletilmesi önerilmektedir.



Şekil 20. En iyi 20 model için nb\_conv\_pool\_layers hiperparametresi sonuçları.

nb\_conv\_pool\_layers parametresi oluşabilecek mimarinin derinliğini büyük ölçüde etkilediği için tanım aralığının artırılması sonrası eğitim için ihtiyaç duyulacak donanım kaynaklarında büyük ölçüde artış oluşabileceği düşünülmektedir.

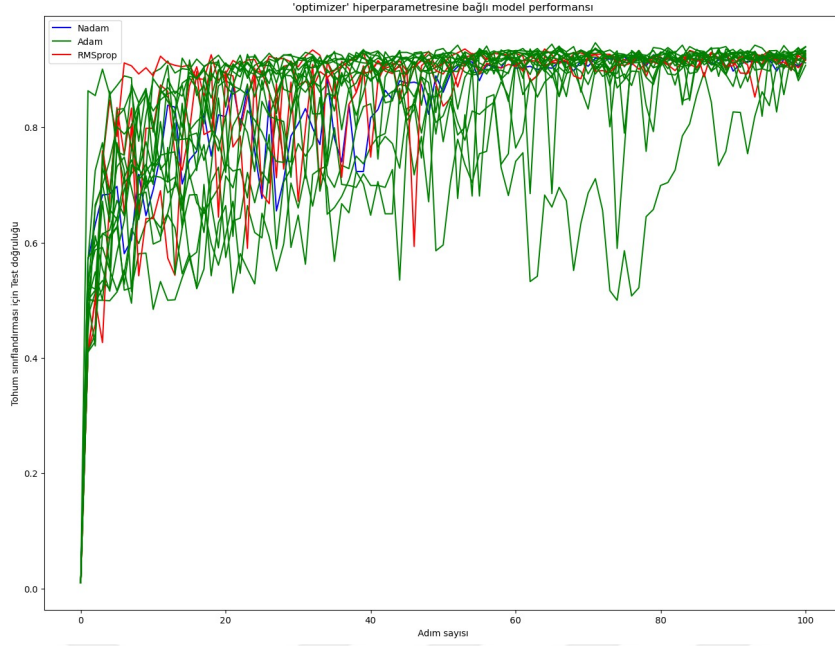
Şekil 21’de one\_more\_fc parametresi için model sonuçları yer almaktadır. Bu parametre sadece 2 farklı model için aktive olmuştur. Kalan 18 modelde değeri null (siyah renk) olarak belirlenmiştir. Parametrenin doğruluk skoruna negatif yönde etki ettiği düşünülmektedir. Parametreden yararlanan iki farklı modelin doğruluk skorları sırasıyla 0,8683 parametresi için %92,83, 1,7511 için %93,25’tir. Tanım aralığı olan 'None' U [0,5488, 1,8221] göz önüne alındığında en iyi 20 modelin üçte ikisinde bu parametreye dair örnekler gelmesi beklenirken sadece 2 defa kullanılması bu parametrenin test doğruluğu açısından negatif yönden etki yarattığını göstermektedir.



Şekil 21. En iyi 20 model için one\_more\_fc hiperparametresi sonuçları.

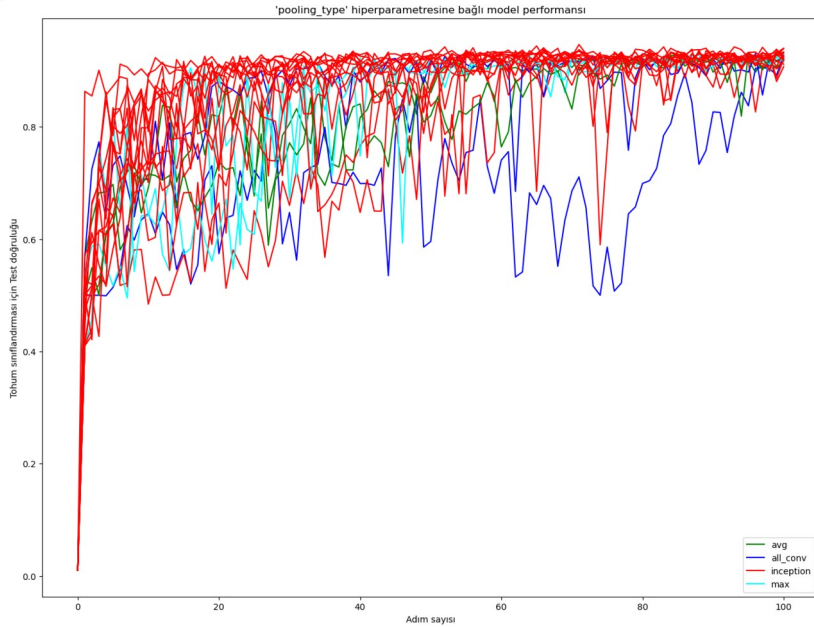
Şekil 22’de optimizasyon parametresi için model sonuçları yer almaktadır. Grafikten görülebileceği üzere en iyi 20 modelin büyük çoğunluğu ADAM optimizasyon algoritması kullanılmaktadır. 1 model NADAM, 3 model RMSprop algoritmasından yararlanmıştır. Sonuçlar ışığında optimizasyon parametresi için sadece Adam olacak şekilde sabitlenmesi yada farklı algoritmaların test edilmesi önerilmektedir.





Şekil 22. En iyi 20 model için optimizer hiperparametresi sonuçları.

Şekil 23’de pooling\_type parametresi için model sonuçları yer almaktadır. Büyük bir üstünlükle inception havuzlama yöntemi en verimli sonuçları elde etmiştir. 20 modelin 2 tanesinde avg, 2 tanesinde all\_conv, 2 tanesinde max havuzlama yöntemi kullanılmıştır.

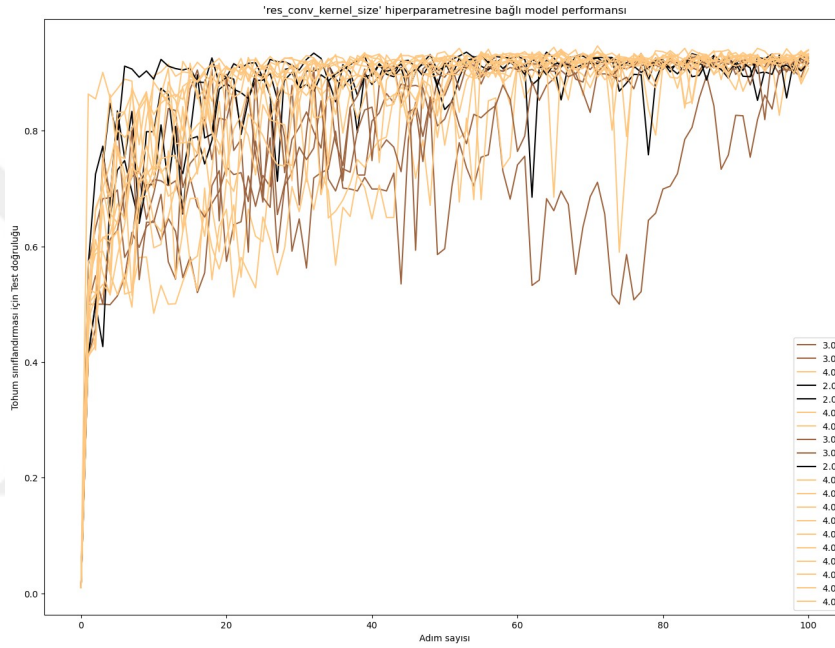


Şekil 23. En iyi 20 model için pooling\_type hiperparametresi sonuçları.

Kalan 14 modelde inception havuzlama yönteminden yararlanılmıştır. Özellikle %94 üzeri test doğruluğu skoruna sahip tüm modellerde inception havuzlama yöntemi kullanıldığı görülmüştür. Sonuçlar ışığında pooling\_type parametresi için

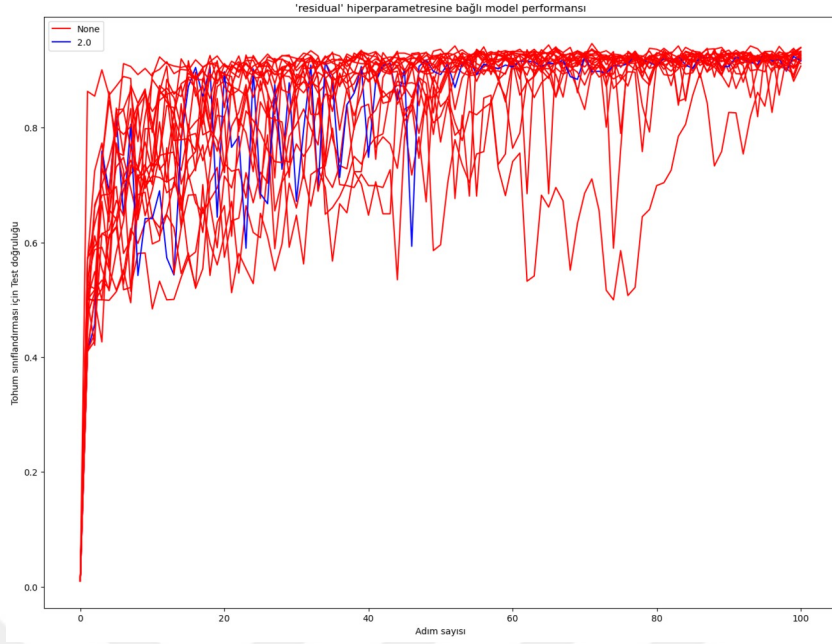
sadece inception olacak şekilde sabitlenmesi yada farklı havuzlama türlerinin test edilmesi önerilmektedir.

Şekil 24'te `res_conv_kernel_size` parametresi için model sonuçları yer almaktadır. Tanım aralığı olan  $[2,4]$  göz önüne alındığında iyi skor elde eden modellerin büyük çoğunluğunda 4 parametresinin kullanıldığı görülmüştür. Sonuçlar ışığında tanım aralığının  $[4,6]$  olacak şekilde yeniden düzenlenmesi önerilmektedir.



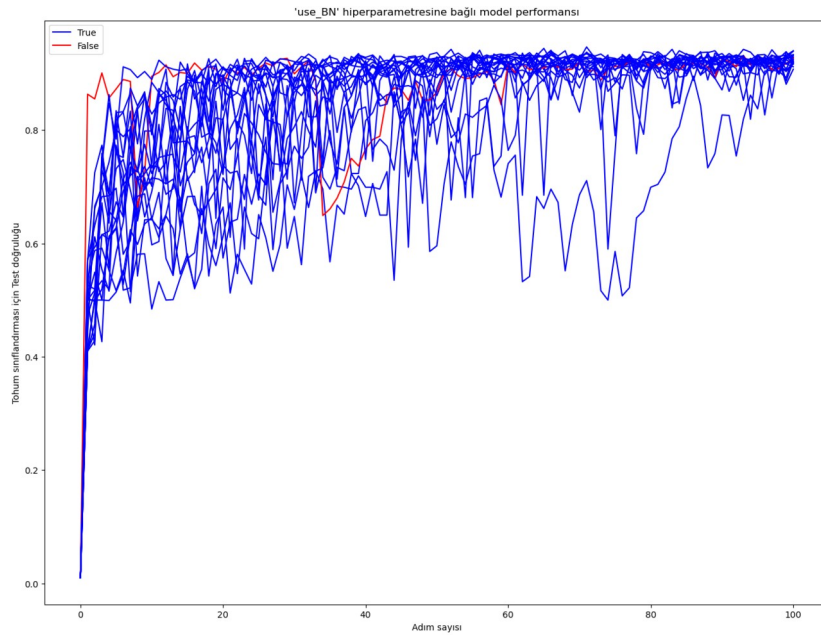
Şekil 24. En iyi 20 model için `res_conv_kernel_size` hiperparametresi sonuçları.

Şekil 25'te residual parametresi için model sonuçları yer almaktadır. Bu parametre sadece bir modelde kullanılmıştır. Parametrenin doğruluk skoruna negatif yönde etki ettiği düşünülmektedir. Artık bağların kullanımının genel anlamda ESA başarımını düşürdüğü görülmüştür. Parametrenin hiperparametre uzayından tamamen çıkarılması önerilmektedir.



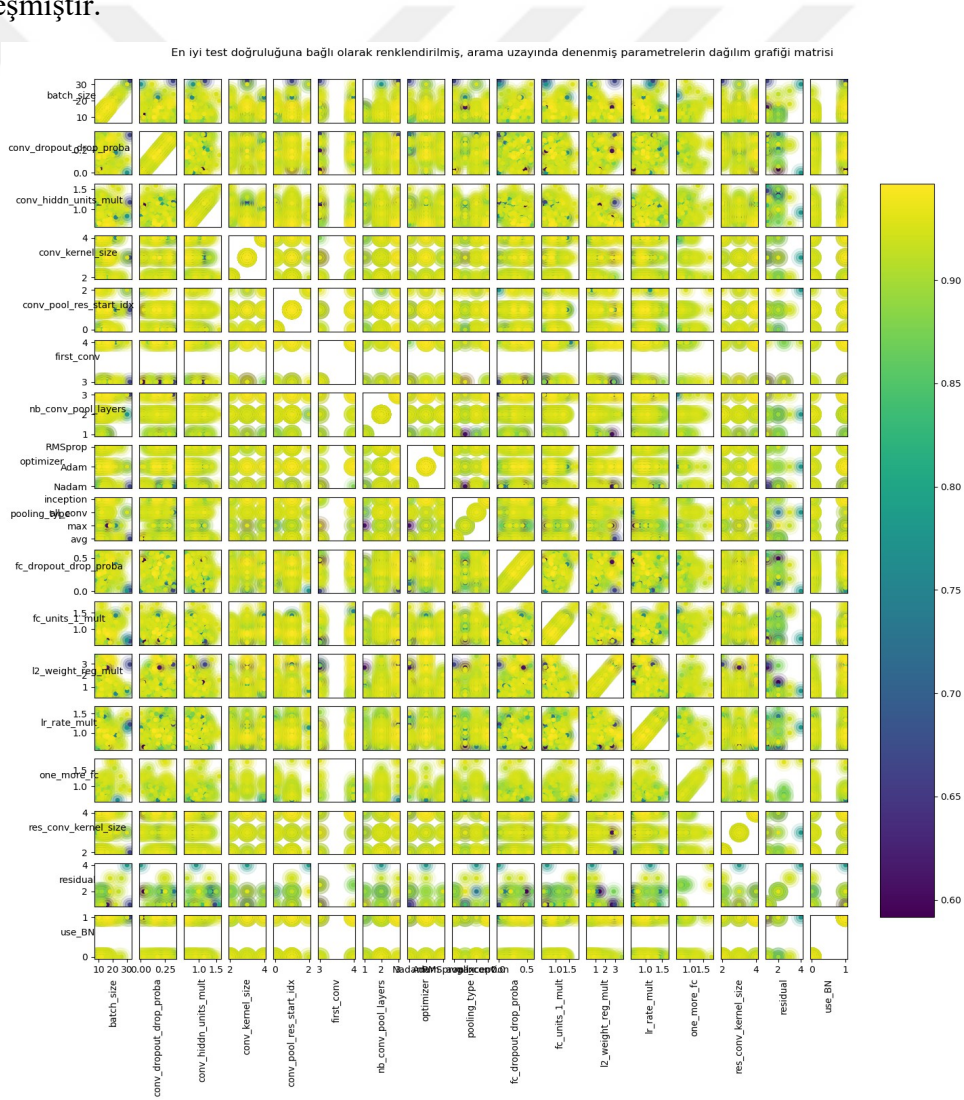
Şekil 25. En iyi 20 model için residual hiperparametresi sonuçları.

Şekil 26’da use\_BN parametresi için model sonuçları yer almaktadır. Bir model hariç modellerin tamamında batch normalizasyonundan yararlanılmıştır. Parametrenin kullanılma ihtimalinin 'True' yapılarak hiperparametre uzayında alacağı değerin sabitlenmesi önerilmektedir.



Şekil 26. En iyi 20 model için use\_BN hiperparametresi sonuçları.

Şekil 27’de modellerin üretilmesi sırasında denenmiş parametrelerin test doğruluğuna bağlı dağılım grafiği yer almaktadır. Bu grafik incelendiğinde batch\_size parametresinin yüksek olmasının skora negatif etkisi olduğu görülmüştür. Tam bağlı katmandaki nöron sayısını belirleyen fc\_units\_1\_mult parametresinin düşük olduğu durumlarda validasyon skorunda benzer şekilde gerileme gerçekleştiği görülmüştür. Özellikle Nadam optimizasyon algoritması ve all\_conv havuzlamanın birlikte kullanıldığı durumlarda test doğrulama skoru büyük ölçüde düşüş göstermiştir. Genel anlamda Nadam optimizasyonunun verimsiz sonuç ürettiği söylenebilir. Özellikle first\_conv parametresinin değerinin 4 yerine 3 olması durumunda negatif yönlü etki görülmüştür. Negatif yönlü başka bir etki ise l2\_weight\_reg\_mult parametresinin 2 değeri üzerinde olması durumunda gerçekleşmiştir.



Şekil 27. Tüm modeller için hiperparametre uzayı üzerinde denenmiş parametrelerin test doğruluğuna bağlı dağılımını gösteren matris grafiği.

Son olarak Şekil 28’de üretilen modeller arasında %94,66 ile en iyi test doğruluğu skorunu elde eden modele ait mimari paylaşılmıştır. Model aktivasyon fonksiyonu olarak gelu, optimizasyon algoritması olarak 'Adam', havuzlama katmanı türü olarak 'inception' kullanmaktadır. Modele ait hiperparametrelerin tümüne aşağıda bulunan Tablo 2.’de yer verilmiştir.

Tablo 2

En iyi modele ait hiperparametre değerleri tablosu

Hiperparametre	Değer
Activation	Gelu
Batch_size	'16.0'
Conv_dropout_drop_proba	'0,19363839453793 '
Conv_hiddn_units_mult	'0,6113404403786434 '
Conv_kernel_size	'3.0 '
Conv_pool_res_start_idx	'1.0 '
Fc_dropout_drop_proba	'0,4118566751213981 '
Fc_units_1_mult	'0,8532731416958078 '
First_conv	4
L2_weight_reg_mult	'3,3371144967559285 '
Lr_rate_mult	'0,9248006083166472 '
Nb_conv_pool_layers	3
One_more_fc	Null
Optimizer	Adam
Pooling_type	Inception
Res_conv_kernel_size	4
Residual	Null
Use_BN	'True'

Tablo 2’de yer alan parametrelerin tamamı daha önceden tanımlanan hiperparametre uzayı içerisinde TPE algoritması yardımıyla otomatik olarak oluşturulmuştur.



piksellik 3 kanaldan (r,g,b) oluşan görüntüleri girdi olarak aldığı görülebilmektedir. Sonraki katman olan 'lambda' üzerinde veri arttırımı işlemi gerçekleştirilmiştir. Modele ait 'first\_conv' hiperparametresi değeri, TPE algoritması tarafından 4 olarak belirlendiği için lambda katmanı sonrası özel evrişim katmanı kullanılmıştır. Bu katmandaki filtre sayısının belirlenmesi için kullanılan 'conv\_hiddn\_units\_mult' hiperparametresinin başlangıç değeri '16' olduğu için bunun yansıması olarak ilk katmanda '16' filtre kullanılmıştır. Bu katmana ait kernel büyüklüğü first\_conv değeri tarafından '4' belirlenmiştir. Stride değeri '1' olarak atanmıştır. Katmanın aktivasyon fonksiyonu 'Gelu' olarak TPE tarafından belirlenmiştir. Mimarinin tamamında aynı aktivasyon fonksiyonundan yararlanılacaktır. Başlangıç 12 ağırlıklarında ilk olarak bu katman üzerinde varsayılan değer (0,0007) ile mimariye ait hiperparametrenin (3,3) çarpımı sonucunda elde edilmiş ve kullanılmıştır. Mimarinin tamamı için yine bu çarpım sonucu oluşan parametreden yararlanılacaktır.

Bir sonraki adımda filtre sayısı varsayılan değer (16) ile 'conv\_hiddn\_units\_mult' parametresinin (0,6113404403786434) döndürdüğü değer çarpımı sonrası yeniden oluşturulmuştur. Bir sonraki katmanın filtre sayısı bu değer tamsayı olan kısmı (9) ile belirlenmektedir. 'nb\_conv\_pool\_layers' parametresi bu mimari için TPE algoritması tarafından '3' olarak atandığı için mimarinin üç farklı evrişim katmanı içermesi zorunluluğu bulunmaktadır. Bu katmanların herhangi birinden sonra artık ağ katmanı yapısının kullanılabilme ihtimali bulunmaktadır. Bu katmanlar sırasıyla 'conv2d\_1', 'conv2d\_8', 'conv2d\_15' katmanlarından oluşmaktadır. Mimariye ait 'Conv\_pool\_res\_start\_idx' hiperparametre değeri '1.0' olduğu için artık ağların kullanılabilmesi için öncesinde en az bir adet evrişimli ağ katmanı kullanılması zorunluluğu oluşmuştur. Bu yüzden incelenen mimari için 'conv2d' katmanı sonrası artık ağ yapısı kullanılabilir. 'use\_BN' parametresi değeri 'True' olduğu için üstte verilen üç farklı katman sonrası batch normalizasyonu katmanı kullanılmıştır.

Artık ağ katmanı yapısı evrişim katmanı sonrası aktive olma şansına sahiptir. Eğer 'use\_BN' parametresi değeri 'True' ise bu aktivasyon batch normalizasyonu katmanı sonrasında gerçekleştirilir. Şekil 28'den görülebileceği üzere üretilen en iyi mimari için artık ağ yapısı, 'batch\_normalization', 'batch\_normalization\_3', 'batch\_normalization\_6'

sonrasında aktive olmuştur. TPE tarafından seçilen havuzlama tipi parametresi tipi inception olduğu için artık ağların oluşturulması sırasında inception azaltma modülü (reduction module) mantığından (Szegedy vd., 2017) yararlanılmıştır. Tez kapsamında uygulanan bu yöntemde bir önceki katmandan gelen girdi öncelikle üç farklı evrişim katmanına aktarılmaktadır. Bu üç evrişim katmanındaki filtreler bir önceki katmana oranla 0.33 ile çarpılıp 1 eklenerek düşürülmektedir. Bu aşamadaki her bir evrişim katmanı kendine özel farklı havuzlama yöntemi ve mimari kullanılan alt bir sinir ağı gibi düşünülebilir. İlk evrişim katmanı, (conv2d\_3, conv2d\_10, conv2d\_17) katmana gelen girdi üzerinde evrişimli havuzlama yöntemi kullanması sonucu oluşan girdiyi iletmektedir. İkinci evrişim katmanı, (conv2d\_4, conv2d\_11, conv2d\_17), kendisine gelen girdiyi filtre sayısı daha yüksek başka bir evrişim katmanına girdi olarak vermek ve sonrasında evrişimli havuzlama yöntemi kullanılmaktadır. Filtre değerinin belirlenmesi için girdi 0.40 ile çarpılıp 1 eklenilir. Üçüncü evrişim katmanı, gelen girdiden yararlanarak max havuzlama yöntemi ile çıktı üretmektedir. Max havuzlama yöntemi için kullanılan havuz büyüklüğü değeri '(3,3)', stride değeri '2' olarak tanımlanmıştır. Son olarak bu üç farklı katmanın çıktısı (concatenate, concatenate\_2, concatenate\_3 ) birleştirilmektedir.

Bu katman sonrası eğer 'Conv\_dropout\_drop\_proba' parametresi değeri sıfır değil ise bu aşamada dropout katmanı aktive olmaktadır. İncelediğimiz mimari için bu değer '0,19363839453793' olduğu için en iyi mimari için bu aşamada dropout katmanı aktive olmuştur. Sonraki adımda artık ağ katmanı yapısı 'nb\_conv\_pool\_layers' parametresi '3' sayısı kadar tekrar gerçekleştirilmektedir. Bu yüzden incelediğimiz mimari için üçüncü tekrar gerçekleştikten sonrası dropout katmanı arkasından tam bağlı katman kullanılmıştır. Tam bağlı katman sayısının bir mi ? Birden çok mu ? Olacağını belirleyen 'One\_more\_fc' parametresi bu mimari için 'Null' olduğundan dolayı sadece bir adet tam bağlı katman kullanılmıştır. Tam bağlı katman sonrası dropout olup, olmamasına karar veren 'Fc\_dropout\_drop\_proba' incelenen mimari için pozitif olduğundan dolayı tam bağlı katman sonrası dropout katmanı yer almaktadır. Son olarak yaptığımız işlem ikili sınıflandırma olduğu için sigmoid aktivasyon fonksiyonu ve hiperparametre uzayı tarafından tanımlanan 12 ağırlıklarından yararlanılarak çıktı katmanı oluşturulmuştur.



## BEŞİNCİ BÖLÜM

### SONUÇ VE ÖNERİLER

Hiperparametreler, günümüzde bir sinir ağının mimarisini belirlemek, eğitim yapılacak ağın özelliklerini tanımlamak yada her ikisini birden gerçekleştirmek (Aszemi ve Dominic, 2019) amacıyla kullanılabilir. Bu tez çalışmasında hem ESA mimarisini belirlemek hemde ağın özelliklerini tanımlamak amacıyla hiperparametre uzayı oluşturulmuştur. Bu çalışmada halkın kullanımına açık haploid ve diploid mısır tohumları görüntülerinden oluşan veri kümesinden yararlanılmıştır. Veri kümesi üzerinde tensorflowa ait `random_flip_left_right` fonksiyonu yardımıyla veri arttırımı (data augmentation) gerçekleştirilmiştir. Çalışmada mısır tohumu sınıflandırması için ESA mimarisi oluşturma amacıyla daha önceden tanımlanmış hiperparametre uzayından yararlanılarak hiperparametre optimizasyonu gerçekleştirilmiş ve 150 farklı model mimarisi oluşturulmuştur. Tezin araştırma konusu eldeki veri kümesi üzerinde verimli bir şekilde sınıflandırma yapabilmek için hiperparametre uzayında bulunan parametrelerin hangilerinin, oluşturulan ESA mimarisine pozitif, hangilerinin negatif etkiye sahip olduğunu araştırmak ve pozitif etkiyi arttırmak, negatif etkiyi azaltmak için gerçekleştirilebilecek işlemleri tartışmaktır. Tez kapsamında, eğitilebilir parametre sayısının belirlenmesine büyük ölçüde katkı sunan tam bağlı katman nöron sayısının belirlenmesini sağlayan hiperparametreler (`fc_units_1_mult`, `one_more_fc`) bilerek düşük tutularak, aynı veri kümesi için literatürde bulunan çalışmalarda kullanılan mimarilerden daha düşük parametre sayısına sahip olmasına rağmen benzer ya da daha iyi doğruluk skoru elde eden mimarilerin üretilip, üretilmeyeceği araştırılmıştır.

Üretilen en iyi modelin mimarisi başarıyı daha önce bir çok kez kanıtlanmış (Altuntaş vd., 2019) VGG-19 modelinden daha iyi test doğruluğu skoru elde etmiştir. Üretilen modelin mimarisi (9 milyon), eğitilebilir parametre sayısı yönünden VGG-19'a (144 milyon) oranla çok daha düşüktür. Bu düşüklükten dolayı elde edilen mimarinin model karmaşıklığı daha az olmuş ve modelin aşırı ezberleme (overfitting) ihtimali düşürülmüştür. Parametre sayısının düşürülmesinin aynı zamanda daha hızlı eğitim ve değerlendirme süreleri elde edilmesine sağladığı için üretilen modelin bu yönden de daha başarılı olduğu söylenebilmektedir. Üretilen modele ait eğitilebilir parametre sayısının

düřüklüğünden kaynaklanan başka bir avantaj ise modelin eğitilebilme ve kullanılabilmesi için gereken gerekli donanımın görece olarak daha az maliyetli olmasıdır. Bu durum üretilen mimarinin uç cihazlarda (edge devices) kullanılabilme olasılığı arttırmaktadır.

Tez kapsamında üretilen tüm modellere ait mimarileri oluşturan parametreler analiz edilmiş bunların arasından en iyi 20 model arasında karşılaştırma gerçekleştirilmiştir. Karşılaştırma sonucunda hiperparametre uzayında ileriye dönük yapılabilecek iyileştirmeler tartışılmıştır. Üretilen en iyi model %94,66 test doğruluğu skoru ile daha önceki çalışmalardan (Altuntaş vd., 2019) (Dönmez, 2020) daha verimli sonuçlar ortaya çıkarmıştır.

Tez çıktılarından yararlanılarak yeni bir hiperparametre uzayı oluşturması ve sonrasında tekrar model üretim işleminin gerçekleştirilmesinin, test doğruluğu açısından daha verimli sonuç üreten yeni modellerin üretimine olanak sağlayacağı düşünülmektedir. Çalışmanın iyileştirilmesi için yapılabilecek başka bir işlem ise başlangıç parametrelerinin değiştirilmesidir. Özellikle tam bağlı katmanlarda bulunan nöron sayılarını belirleyen hiperparametrelere (fc\_units\_1\_mult, one\_more\_fc) ait varsayılan başlangıç değerlerinin (200, 300) değiştirilmesi sonucu test doğruluk skoru üzerinde iyileşme görülmesi beklenmektedir. Gelecekte yapılabilecek başka bir iyileştirme ise üretilen mimari, mimarilerin farklı veri kümeleri üzerinde test edilmesidir. Üretilen mimarilerin her tür veri kümesinde verimli sonuçlar elde edip etmediği tez kapsamında incelenmemiştir. Bu incelemenin gerçekleştirilmesi üretilen model mimarilerinin yeni veri kümelerine verimli bir şekilde adapte olup olamayacağını gösterecektir.

Derin öğrenme ve hiperparametre optimizasyonu için gerekli olan donanım maliyetinin günümüzde gittikçe düşmesi ile bu konulara olan ilginin artması beklenmektedir. Bu ilginin artması sonucunda tüm veri kümeleri üzerinde görece olarak verimli sonuç elde eden VGG-19, AlexNet, GoogleNet, ResNet ve benzerleri gibi mimarilerin yerine, hiperparametre optimizasyonundan yararlanılarak ele alınan veri kümesine özel üretilmiş mimarilerin daha sık kullanılacağı öngörülmektedir.

## KAYNAKÇA

- Altuntaş, Y., Cömert, Z., Kocamaz, A. F. (2019). "Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach". *Computers and Electronics in Agriculture*, Cilt 163, s. 104874, ISSN: 0168-1699, DOI: <https://doi.org/10.1016/j.compag.2019.104874>, URL: <https://www.sciencedirect.com/science/article/pii/S0168169919300481>.
- Altuntaş, Y., Kocamaz, A. F., Cengiz, R., Esmeray, M. (2018). "Classification of haploid and diploid maize seeds by using image processing techniques and support vector machines". içinde: *2018 26th Signal Processing and Communications Applications Conference (SIU)*, ss. 1–4, DOI: 10.1109/SIU.2018.8404800.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., Farhan, L. (2021). "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions". *Journal of big Data*, Cilt 8, ss. 1–74.
- Aszemi, N. M., Dominic, P. (2019). "Hyperparameter optimization in convolutional neural network using genetic algorithms". *International Journal of Advanced Computer Science and Applications*, Cilt 10, Sayı 6.
- Basha, S. S., Dubey, S. R., Pulabaigari, V., Mukherjee, S. (2020). "Impact of fully connected layers on performance of convolutional neural networks for image classification". *Neurocomputing*, Cilt 378, ss. 112–119.
- Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B. (2011). "Algorithms for hyper-parameter optimization". *Advances in neural information processing systems*, Cilt 24.
- Bishop, C. M., Nasrabadi, N. M. (2006). "Pattern recognition and machine learning", c. 4, 4, Springer.
- Bochinski, E., Senst, T., Sikora, T. (2017). "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms". içinde: *2017 IEEE international conference on image processing (ICIP)*, IEEE, ss. 3924–3928.

- Boopathi, N. M. (2020). "Marker-Assisted Selection (MAS)". içinde: Genetic Mapping and Marker Assisted Selection: Basics, Practice and Benefits, Singapore: Springer Singapore, ss. 343–388, ISBN: 978-981-15-2949-8, DOI: 10.1007/978-981-15-2949-8\_9, URL: [https://doi.org/10.1007/978-981-15-2949-8\\_9](https://doi.org/10.1007/978-981-15-2949-8_9).
- Boote, B. W., Freppon, D. J., De La Fuente, G. N., Lübberstedt, T., Nikolau, B. J., Smith, E. A. (2016). "Haploid differentiation in maize kernels based on fluorescence imaging". *Plant Breeding*, Cilt 135, Sayı 4, ss. 439–445.
- Carena, M., Hallauer, A., Filho, J. (2010). "Quantitative Genetics in Maize Breeding". *Huamana Press*, ISBN: 978-1-4419-0765-3, DOI: 10.1007/978-1-4419-0766-0.
- Chaikam, V., Molenaar, W., Melchinger, A. E., Boddupalli, P. M. Ara. (2019). "Doubled haploid technology for line development in maize: technical advances and prospects". *Theoretical and Applied Genetics*, Cilt 132, Sayı 12, ss. 3227–3243, ISSN: 1432-2242, DOI: 10.1007/s00122-019-03433-x, URL: <https://doi.org/10.1007/s00122-019-03433-x>.
- Chaikam, V., Nair, S. K., Babu, R., Martinez, L., Tejomurtula, J., Boddupalli, P. M. (2015). "Analysis of effectiveness of R1-nj anthocyanin marker for in vivo haploid identification in maize and molecular markers for predicting the inhibition of R1-nj expression". *Theoretical and applied genetics*, Cilt 128, ss. 159–171.
- Bayhan, M., Ozkan, R., Albayrak, O., Yıldırım, M., Akıncı, C. (2021). "İn Vivo Double Haploid Tekniği ile Yerel Çeşitlerden Elde Edilen Haploid Bitkilerin Saf Hat Olarak Kullanılabilirliğinin Belirlenmesi". *Kahramanmaraş Sütçü İmam Üniversitesi Tarım ve Doğa Dergisi*, Cilt 24, Sayı 5, ss. 1029–1036, DOI: 10.18016/ksutarimdog.vi.825121.
- Chaikam V., Gowda M., Martinez L., Ochieng J., Omar H.A., Prasanna B.M. (2020). "Improving the Efficiency of Colchicine-Based Chromosomal Doubling of Maize Haploids". *Plants*, 9(4):459. DOI:10.3390/plants9040459
- Cleophas, T. J., Zwinderman, A. H., Cleophas-Allers, H. I. (2013). "Machine learning in medicine". c. 9, Springer.

- Coelho, I. M., Coelho, V. N., Luz, E. J. d. S., Ochi, L. S., Guimaraes, F. G., Rios, E. (2017). "A GPU deep learning metaheuristic based model for time series forecasting". *Applied Energy*, Cilt 201, ss. 412–418.
- Couto, E., Davide, L., Bustamante, F., Pinho, R., Silva, T. (2013). "Identification of haploid maize by flow cytometry, morphological and molecular markers". *Ciência e Agrotecnologia*, Cilt 37, ss. 25–31, DOI: 10.1590/S1413-70542013000100003.
- Deng, L., Yu, D. (2014). "Deep learning: methods and applications". *Foundations and trends® in signal processing*, Cilt 7, Sayı 3–4, ss. 197–387.
- Dozat, T. (2016). "Incorporating nesterov momentum into adam" .
- Dowswell, C. (2019), *Maize in the third world*, CRC press.
- Dönmez, E. (2020). "Classification of haploid and diploid maize seeds based on pre-trained convolutional neural networks". *Celal Bayar University Journal of Science*, Cilt 16, Sayı 3, ss. 323–331.
- El Naqa, I., Murphy, M. J. (2015). "What is machine learning". Springer, ss. 4–5.
- Fielding, A. (1999). *Machine learning methods for ecological applications*, Springer Science & Business Media.
- Ford, R. H. (2000). "Inheritance of kernel color in corn: explanations & investigations". *The American Biology Teacher*, ss. 181–188.
- gamete (2020). *Encyclopedia Britannica*, Britannica, Erişim adresi: <https://www.britannica.com/science/gamete>
- Garbin, C., Zhu, X., Marques, O. (2020). "Dropout vs. batch normalization: an empirical study of their impact to deep learning". *Multimedia Tools and Applications*, Cilt 79, ss. 12777– 12815.
- Greis, N. P., Nogueira, M. L., Bhattacharya, S., & Schmitz, T. (2020). "Physics-guided machine learning for self-aware machining". *AAAI Spring Symposium on AI and Manufacturing*.
- guillaume-chevalier (2015). *Hyperopt-Keras-CNN-CIFAR-100*, URL: <https://github.com/guillaume-chevalier/Hyperopt-Keras-CNN-CIFAR-100> (erişim tarihi: 20/05/2023).

- Gyorfi, L., Ottucsak, G., Walk, H. (2012). "Machine learning for financial engineering". c. 8, World Scientific.
- Hallauer, A. R., Carena, M. J. (2009). "Maize". içinde: *Cereals*, New York, NY: Springer US, ss. 3–98, ISBN: 978-0-387-72297-9, DOI: 10 . 1007 / 978 - 0 - 387 - 72297 - 9 \_ 1, URL: [https://doi.org/10.1007/978-0-387-72297-9\\_1](https://doi.org/10.1007/978-0-387-72297-9_1).
- Hendrycks, D., & Gimpel, K. (2016). "Gaussian error linear units (gelus)". *arXiv preprint arXiv:1606.08415*.
- Hinton, G. E., Osindero, S., Teh, Y.-W. (2006). "A fast learning algorithm for deep belief nets". *Neural computation*, Cilt 18, Sayı 7, ss. 1527–1554.
- homozygote (2012). içinde: *Encyclopedia Britannica*, Britannica, Erişim adresi: <https://www.britannica.com/science/homozygote>.
- Huang, J. (2020). RMSProp, Erişim adresi: <https://optimization.cbe.cornell.edu/index.php?title=RMSProp>
- Hyperopt (2023). Defining a Search Space, URL: [http://hyperopt.github.io/hyperopt/getting-started/search\\_spaces/](http://hyperopt.github.io/hyperopt/getting-started/search_spaces/) (erişim tarihi: 20/06/2023).
- Ioffe, S., Szegedy, C. (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". içinde: ss. 448–456, URL: <http://jmlr.org/proceedings/papers/v37/lofffe15.pdf>.
- Kingma, D. P., Ba, J. (2017). "Adam: A Method for Stochastic Optimization", arXiv: 1412.6980
- LeCun, Y., Bengio, Y., Hinton, G. (2015). "Deep learning". *Nature*, Cilt 521, Sayı 7553, ss. 436–444.
- Lin, J., Yu, L., Li, W., Qin, H. (2018). "Method for identifying maize haploid seeds by applying diffuse transmission near-infrared spectroscopy". *Applied Spectroscopy*, Cilt 72, Sayı 4, ss. 611–617.
- Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). "On the variance of the adaptive learning rate and beyond"., Doi: 10.48550/arXiv.1908.03265

- Ma, J., & Yarats, D. (2021). "On the adequacy of untuned warmup for adaptive optimization. "In *Proceedings of the AAAI Conference on Artificial Intelligence*, Cilt 35, Sayı 10, ss. 8828-8836).
- Melchinger, A. E., Schipprack, W., Mi, X., Mirdita, V. (2015). "Oil content is superior to oil mass for identification of haploid seeds in maize produced with high-oil inducers". *Crop Science*, Cilt 55, Sayı 1, ss. 188–195.
- Melchinger, A. E., Schipprack, W., Würschum, T., Chen, S., Technow, F. (2013). "Rapid and accurate identification of in vivo-induced haploid seeds based on oil content in maize". *Scientific reports*, Cilt 3, Sayı 1, s. 2129.
- Mitchell, T. (1997). *Machine Learning*, McGraw-Hill International Editions, McGraw-Hill, ISBN: 9780071154673, URL: <https://books.google.com.tr/books?id=EoYBngEACAAJ>.
- O'Shea, K., Nash, R. (2015). "An introduction to convolutional neural networks". arXiv preprint arXiv:1511.08458.
- Park, S., Kwak, N. (2017). "Analysis on the Dropout Effect in Convolutional Neural Networks". içinde: *Computer Vision – ACCV 2016*, ed. S.-H. Lai, V. Lepetit, K. Nishino, Y. Sato, Cham: Springer International Publishing, ss. 189–204, ISBN: 978-3-319-54184-6.
- prakharr0y (2020). Intuition of Adam Optimizer, 07.06.2023 tarihinde erişilmiştir, URL: <https://www.geeksforgeeks.org/intuition-of-adam-optimizer>.
- Prasanna, B., Chaikam, V., Mahuku, G. (2012). Doubled haploid technology in maize breeding: theory and practice, *Cimmyt*.
- Prigge, V., Schipprack, W., Mahuku, G., Atlin, G. N., Melchinger, A. E. (2012). "Development of in vivo haploid inducers for tropical maize breeding programs". *Euphytica*, Cilt 185, ss. 481–490.
- Richter, M. L., Byttner, W., Krumnack, U., Wiedenroth, A., Schallner, L., & Shenk, J. (2021). "(Input) size matters for CNN classifiers". *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Proceedings, Part II 30*(ss. 133-144). Springer International Publishing.

- Sanders, S., Giraud-Carrier, C. (2017). "Informing the Use of Hyperparameter Optimization Through Metalearning". içinde: *2017 IEEE International Conference on Data Mining (ICDM)*, ss. 1051–1056, DOI: 10.1109/ICDM.2017.137.
- Sarker, I. H. (2021), "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions". *SN Computer Science*, Cilt 2, Sayı 6, s. 420.
- Segui-Simarro, J. M., Jacquier, N. M. A., Widiez, T. (2021). "Overview of In Vitro and In Vivo Doubled Haploid Technologies". içinde: *DOUBLED HAPLOID TECHNOLOGY, VOL. 1: General Topics, Alliaceae, Cereals*, ed. J. Segui-Simarro, c. 2287, *Methods in Molecular Biology*, Humana Press, ss. 3–22, ISBN: 978-1-0716-1315-3; 978-1-0716-1314- 6, DOI: 10.1007/978-1-0716-1315-3\_1.
- Simonyan, K., Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition". arXiv preprint arXiv:1409.1556.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., Riedmiller, M. (2014). "Striving for simplicity: The all convolutional net". arXiv preprint arXiv:1412.6806.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting". *The journal of machine learning research*, Cilt 15, Sayı 1, ss. 1929–1958.
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". içinde: *Proceedings of the AAAI conference on artificial intelligence*, c. 31, 1.
- Taylor, J., Chiou, C.-P., Bond, L. J. (2019). "A methodology for sorting haploid and diploid corn seed using terahertz time domain spectroscopy and machine learning". içinde: *AIP Conference Proceedings*, c. 2102, 1, AIP Publishing.
- Veeramani, B., Raymond, J. W., Chanda, P. (2018). "DeepSort: deep convolutional networks for sorting haploid maize seeds". *BMC bioinformatics*, Cilt 19, ss. 1–9.
- Wang, Y., Lv, Y., Liu, H., Wei, Y., Zhang, J., An, D., Wu, J. (2018). "Identification of maize haploid kernels based on hyperspectral imaging technology". *Computers and Electronics in Agriculture*, Cilt 153, ss. 188–195, ISSN: 0168-1699, DOI: <https://doi.org/10.1016/j.compag.2018.08.012>, URL: <https://www.sciencedirect.com/science/article/pii/S0168169918302953>.



- Watanabe, S. (2023). "Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance". arXiv: 2304.11127 [cs.LG].
- WorldBank (2022). Key Highlights: Country Climate and Development Report for Türkiye, 06.05.2023 tarihinde erişilmiştir, URL: <https://www.worldbank.org/en/country/turkey/brief/key-highlights-country-climate-and-development-report-for-turkiye>.
- Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C. (2018). "Machine learning and deep learning methods for cybersecurity". Ieee access, Cilt 6, ss. 35365–35381.
- Yu, L., Liu, W., Li, W., Qin, H., Xu, J., Zuo, M. (2018). "Non-destructive identification of maize haploid seeds using nonlinear analysis method based on their near-infrared spectra". *Biosystems engineering*, Cilt 172, ss. 144–153.