

GAZİ

JOURNAL OF ENGINEERING SCIENCES

Hyper Parameter Analysis in Recognition of Handwritten Digits Using Convolutional Neural Network

Tuncay Yiğit^a, Şerafettin Atmaca^b, Remzi Gürfidan^c, Recep Çolak^d

Submitted: 28.11.23 Revised: 18.12.2023 Accepted: 19.12.2023 doi:10.30855/gmbd.0705S25

ABSTRACT

Keywords: Convolutional Neural Network, Hyperparameter Tuning, Deep Learning

^{a,*} Süleyman Demirel University,
Faculty Of Engineering,
Dept. of Computer Engineering
32200 - Isparta, Türkiye
Orcid: 0000-0001-7397-7224
e mail: tuncayyigit@sdu.edu.tr

^b Isparta University Of Applied Science,
Dept. of Information Technology
32200 - Isparta, Türkiye
Orcid: 0000-0003-2407-1113

^c Isparta University Of Applied Science,
Yalvaç Technical Sciences Vocational
School
32200 - Isparta, Türkiye
Orcid: 0000-0002-4899-2219

^d Isparta University Of Applied Science,
Distance Education Vocational School
32200 - Isparta, Türkiye
Orcid: 0000-0002-7119-6202

*Corresponding author:
serafettinatmaca@isparta.edu.tr

Anahtar Kelimeler: Evrişimli Sinir Ağı, Hiper parametre, Derin Öğrenme

Recognition of handwritten digits has recently gained importance and attracted the attention of many scientists, as it is used in many machine learning, deep learning and computer vision applications. Hyperparameter optimization involves determining a set of values aimed at increasing accuracy in both classification and prediction. It is also aimed to optimize the performance in feature selection by regulating the parameters selected by the algorithms more accurately. In this study, a convolutional neural network was used to recognize handwritten digits using the MNIST dataset. There are many open source hyperparameter libraries that deep learning developers can use to determine hyperparameters. In the developed model, hyperparameter optimization techniques were applied using Optuna, HyperOpt and Scikit-optimize libraries and the results were evaluated. Optimization times for hyperparameter libraries and the change in the success rate in recognizing handwritten digits were analyzed. The model trained with randomly given parameters achieved 78.45%, 97.13%, 75.62%, 76.95%, 97.46% and 97.27% accuracy, while the model trained with optimized hyperparameters achieved 99.26% accuracy.

Evrişimli Sinir Ağı Kullanarak El yazısı Rakamların Tanınmasında Hiper Parametre Analizi

ÖZ

El yazısı rakamların tanınması birçok makine öğrenimi, derin öğrenme ve bilgisayarla görme uygulamalarında kullanıldığından, son zamanlarda önem kazanmış ve birçok bilim insanının ilgisini çekmiştir. Hiper parametre optimizasyonu, hem sınıflandırma hem de tahmin etmede doğruluğu artırmayı amaçlayan bir dizi değerlerin belirlenmesini içermektedir. Ayrıca algoritmalar tarafından seçilen parametreler daha doğru şekilde düzenlenerek özellik seçiminde performansın optimize edilmesi de amaçlanmaktadır. Bu çalışmada MNIST veri seti kullanılarak el yazısı ile yazılmış rakamların tanınmasında evrişimli sinir ağı kullanılmıştır. Hiper parametre belirlemede derin öğrenme geliştiricilerinin kullanabileceği birçok açık kaynaklı hiper parametre kütüphanesi mevcuttur. Geliştirilen modelde Optuna, HyperOpt ve Scikit-optimize kütüphaneleri kullanılarak hiper parametre optimizasyon tekniklerinin uygulaması yapılmış ve sonuçlar değerlendirilmiştir. Hiper parametre kütüphaneleri için optimizasyon süreleri, el yazısı rakamların tanınmasında başarı oranındaki değişim analiz edilmiştir. Rastgele verilmiş parametrelerle eğitilen model %78,45, %97,13, %75,62, %76,95, %97,46 ve %97,27 doğruluk elde ederken, optimize edilmiş hiper parametrelerle eğitilen model ile %99,26 doğruluk elde etmiştir.

1. Giriş (Introduction)

Günümüzde, Yapay Zeka (YZ) bir çok alanda yaygın bir şekilde kullanılır hale gelmiştir. Teknolojide yaşanan gelişmelerle birlikte bilgisayarların işlem gücündeki artış, büyük miktardaki verilerin analiz edilmesini hızlandırmıştır. Yapay Zeka genel olarak insan zekasının yapabileceği işlemleri, bilgisayar teknolojileri ile yapılması olarak tanımlanmıştır [1]. Son yıllarda çeşitli denetimli ve denetimsiz öğrenme problemlerinin üstesinden gelmek için derin öğrenmeye dayalı yöntemlere büyük ilgi gösterildi. En yaygın kullanılan ve iyi bilinen yaklaşımlar arasında, girdi verilerinden önemli özellikleri otomatik olarak çıkarabilen bir tür sinir ağı olan evrişimli sinir ağları (CNN'ler) yer almaktadır. Evrişimli sinir ağlarının özellikleri, özellikle de çok boyutlu girdilerden özelliklerin çıkarılmasındaki yeterlilikleri, onları bilgisayarlı görme alanındaki sorunları çözmek için cazip bir seçenek haline getirmektedir [2]. Örüntü tanıma alanındaki önemli araştırma merkezlerinden biri de MNIST(Değiştirilmiş Ulusal Standartlar ve Teknoloji Enstitüsü)'tir. MNIST el yazısı rakam veritabanı, mükemmel araştırma ve pratik uygulamaları nedeniyle örüntü tanıma araştırmalarında büyük önem taşımaktadır. MNIST veri kümesi, her biri 28 x 28 piksel boyutunda olan ve makine öğrenimi modellerinin eğitimi ve test edilmesi için özel olarak oluşturulmuş 70.000 el yazısı rakamdan oluşur. Bu araştırma alanındaki en önemli gelişmelerden biri MNIST veri seti ve rakam tanıma için evrişimli sinir ağlarının (CNN) geliştirilmesidir [3].

Derin öğrenme algoritmaları genellikle eğitim süresi ve doğruluk gibi performans faktörlerini önemli ölçüde etkileyebilen kullanıcı tanımlı belirli hiper parametre değerlerine ihtiyaç duyar [4]. Hiper parametreler, ağ yapısının ve eğitim sürecinin çeşitli özelliklerinin değerlerini belirler. Hiper parametre örnekleri arasında öğrenme oranı, katman sayısı, gizli katmanlar, filtreler, bırakma oranı, yineleme sayısı, yığın boyutu, aktivasyon işlevi, optimize edici ve düzenleme yer alır. Bu parametrelerin ayarlanması, bir derin öğrenme modelinin geliştirilmesinde önemli bir adımdır. Hiper parametre ayarı manuel veya otomatik olarak yapılabilir. Ancak değerlerin manuel olarak seçilmesi verimli bir yöntem değildir ve daha fazla zaman alabilir [5]. Hiper parametre optimizasyonun temel amacı, hiper parametre ayarlama sürecini otomatikleştirerek kullanıcıların öğrenme modellerini gerçek sorunlara verimli bir şekilde uygulamasını sağlamaktır [6].

Bu çalışmada, öncelikle MNIST veri seti ile el yazısı rakamların evrişimli sinir ağı ile tanınması açıklanmış, derin öğrenme uygulamalarında kullanılan açık kaynak hiper parametre optimizasyon çerçeveleri incelenmiş ve Optuna, HyperOpt, Scikit-optimize kütüphaneleri kullanılarak hiper parametre optimizasyon tekniklerinin uygulaması yapılmıştır.

2. Literatür Taraması (Literature Review)

Bergstra vd. yapmış oldukları çalışmada, farklı hiper parametre optimizasyon algoritmalarını incelemektedir. Izgara arama, rastgele arama, Bayes optimizasyonu ve sıralı model tabanlı optimizasyon (SMBO) yöntemleri üzerinde durulmaktadır. Algoritmanın güçlü yönleri, zayıf yönleri ve kullanım senaryoları hakkında bilgiler sunulur [7]. Lindauer vd. Bayes Optimizasyonu için güçlü ve uyarlanabilir bir sistem olan SMAC3'ü geliştirmişlerdir. Bu model, kullanıcıların minimum sayıda yapılandırmayı değerlendirerek performansını hızlı bir şekilde artırmaya olanak tanıyan ve algoritmaları, veri kümeleri ve uygulamaları için en iyi sonuçları veren hiper parametre ayarlarını etkili bir şekilde belirlemelerine olanak tanır. [8]. Liv vd. yapmış oldukları çalışmada, bandit tabanlı bir hiper parametre optimizasyon yöntemi olan Hyperband üzerine bir çalışma yapmışlardır. Hyperband, bir dizi rasgele yapılandırmayı hızla ortadan kaldırarak en iyi performans gösteren hiper parametreleri bulmayı amaçlar. Makale, farklı uygulama senaryolarında Hyperband yaklaşımının etkinliğini, verimliliğini ve performansını tartışmaktadır [9]. Chooi vd. el yazısı karakter tanıma için evrişimli sinir ağına dayalı bir algoritma geliştirmeyi amaçlamışlardır. Oluşturulan modelde her karakter, saf özellik çıkarım sistemine girdi görevi gören ikili değerlerle tanımlanmıştır. Önerilen metod ile karakterlerin tanıma oranını %88,70 doğrulukla tespit edilmiştir. [10]. Khanday vd. yapmış oldukları çalışmada CIFAR10 ve FashionMNIST veri setlerini kullanarak görüntü sınıflandırmada filtre boyutunun hiper parametresi optimize etmişlerdir. Çeşitli filtre boyutlarına göre modellerin doğruluğu üzerindeki etkilerini analiz etmişlerdir. Deneyin sonuçları, filtre boyutu arttıkça doğruluğun azaldığını göstermektedir [11]. Seng vd, araştırmalarında MNIST veri setini analiz etmek için fast.ai'yi kullanmışlardır. Bu analizin odak noktası, CNN ResNet-18 modelini el yazısı rakamları doğru bir şekilde tanıyacak şekilde eğitmektir. Çalışmada GoogLeNet, ResNet-50, ResNeXt-50, MobileNet v2 ve Wide ResNet-50 olmak üzere beş farklı model kullanılmış ve nihai amaç, el yazısı rakamların tanınmasında hangi mimarinin en yüksek hassasiyeti sağladığını belirlemektir. Ayrıca araştırmacılar, beş modelin tamamı için eğitim süresini, en üst sıradaki hata oranlarını, ilk beş hata

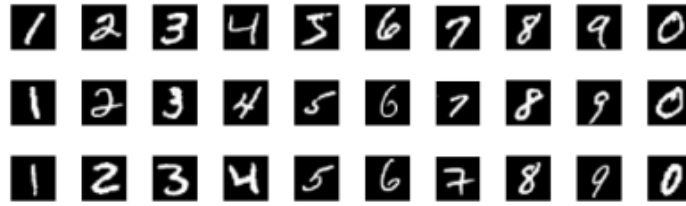
oranını ve model boyutlarını karşılaştırmışlardır. [12]. Shao vd. eğitimde hızlı yakınsama oranıyla MNIST el yazısı rakam veri setinin tanınma oranını daha da artırmak için bir CNN modeli geliştirmiştir. Ortaya konulan model, özellik çıkarımını kolaylaştıran üç evrişim ve aktivasyon katmanından oluşan çok katmanlı bir derin öğrenme yapısını içermektedir. Ayrıca sınıflandırmayı mümkün kılmak için iki adet tamamen bağlantılı katman dahil edilmiştir. Tanıma performansını arttırmak için modelin hiper parametreleri optimize edilmiştir. Bu hiper parametreler toplu büyüklükleri, çekirdek boyutlarını, toplu normalleştirmeyi, aktivasyon fonksiyonunu ve öğrenme oranını içerir [13]. Bischl ve arkadaşları çalışmalarında, hiper parametre performans değerlendirme, hiper parametre optimizasyonunun makine öğrenimi işlem hatlarıyla nasıl birleştirileceği, çalışma zamanı iyileştirmeleri ve paralelleştirme de dahil olmak üzere hiper parametre optimizasyonu yürütülürken yapılması gereken önemli seçimler hakkında pratik tavsiyelerde bulunmuşlardır. [14]. Baldominos vd. bu çalışmada, CNN topolojilerinin tasarımını otomatikleştirmek için nöroevrimden yararlanmışlardır. Bunu başarmak için birleşik bir çerçeve oluşturmuşlar ve genetik algoritmalara ve gramer evrimine dayanan iki yenilikçi yaklaşım tasarlamışlardır. [15].

3. Meteryal ve Metot (Material and Method)

Bu bölümde el yazısı rakamların tanınmasında kullanılan MNIST veri seti, kullanılan evrişimli sinir ağı modeli, hiper parametre optimizasyonu ve açık kaynak hiper parametre kütüphaneleri incelenmiştir.

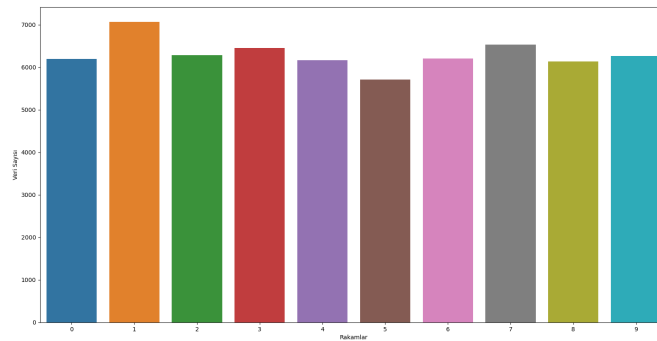
3.1. Veri seti (Data set)

MNIST veri tabanı, rakamların tanınması için 0 ile 9 arasında değişen rakamlardan 60000 eğitim ve 10000 test verisi içermektedir. Her rakam 28*28 boyutunda ve 784 piksellik gri seviyeli bir görüntüdür [16]. Veri setine ait bazı örnekler Şekil 1’de verilmiştir.



Şekil 1. MNIST veri seti örnekleri [17] (MNIST dataset examples)

Rakamlara ait verinin her pikseli, o pikselin parlaklığını veya koyuluğunu gösteren tek bir piksel değeriyle ilişkilendirilir; daha yüksek sayılar daha koyuyu gösterir. Bu piksel değeri 0 ile 255 arasında bir tamsayıdır. 0 ile 9 arasındaki rakamlara ait veri sayıları Şekil 2’de verilmiştir.



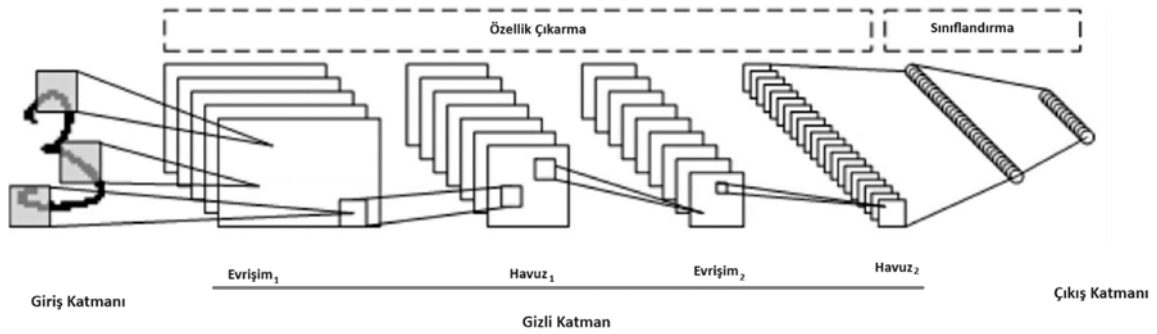
Şekil 2. Veri büyüklükleri (Data sizes)

MNIST, el yazısı rakamları tanımayı amaçlayan modellerin geliştirilmesi ve değerlendirilmesi için önemli bir veri kümesidir. Bu veri seti, araştırmacılara ve uygulayıcılara çeşitli rakam tanıma modellerinin performansını ve insan el yazısındaki varyasyonları ele alma kapasitelerini değerlendirme fırsatı sunar. MNIST, yıllar boyunca bilgisayarla görme ve makine öğrenimi alanında bir unsur olmuştur. Bu veri kümesi kullanılarak dijital görüntü tanıma için çok sayıda teknik geliştirilmiş ve değerlendirilmiştir. MNIST, çeşitli rakam tanıma

modellerini denemek ve karşılaştırmak için sağlam bir platform görevi görmektedir. Ayrıca, rakam görüntü tanıma alanındaki yeni kavramları, metodolojileri ve algoritmaları keşfetmek için bir temel oluşturmaktadır.

3.2. Evrişimli Sinir Ağı (Convolutional Neural Network)

Derin öğrenme, sınıflandırma, tanıma ve algılama gibi alanlarda karmaşık, doğrusal olmayan problemlerin üstesinden gelmek için oldukça etkili bir yöntem haline gelmiştir. Araştırmalarda yaygın olarak kullanılan derin öğrenme, birden fazla katmana sahip bir veya daha fazla yapay sinir ağı modeline dayanır [18]. Hayvanlardaki görsel korteksinin organizasyonuna dayanan [19] Evrişimli Sinir Ağı (CNN), görüntüler gibi ızgara benzeri bir yapıya sahip verileri analiz etmek için özel olarak oluşturulmuş bir derin öğrenme modelidir [20]. Bu yaklaşım 1989 yılında LeCun ve meslektaşları tarafından rapor edilmiş ve bilgisayarla görmede etkili sonuçlar göstermiştir [21]. Bir evrişimli sinir ağı giriş katmanı, çıkış katmanı ve çoklu gizli katmandan oluşur. Gizli katman, yinelemeli evrişim ve havuzlama katmanlarından oluşan bir ağ oluşturur ve sonuçta bir veya daha fazla tam bağlantılı katman elde edilir [22]. CNN'in mimarisi aşağıda açıklanmış ve Şekil 3'de gösterilmiştir.



Şekil 3. MNIST veri seti için CNN yapısına genel bakış (Overview of CNN structure for MNIST dataset)

3.2.1. Giriş katmanı (Input layer)

Evrişimli Sinir Ağı'nın ilk katmanı olan bu bölüm ağa ham veri girişini gerçekleştirir ve görüntünün piksel değerlerini tutar [23]. Bu katmandaki veri boyutunun önemi, modelin performansını optimize etmenin anahtarıdır. Daha yüksek giriş görüntü boyutları daha iyi görüntü seçim seçenekleri sunarken, aynı zamanda test süresini uzatabilecek ve ağ performansını artırabilecek yüksek bellek ve eğitim süresi gereksinimleri de taşır. Giriş görüntüsü boyutu için daha az sayıda düğüm seçmek bellek gereksinimlerini azaltır ve eğitim süresini kısaltır [24]. Bu sebeple hazırlanan CNN modelinde giriş katmanına 28x28 çözünürlüğünde görseller sunulmaktadır.

3.2.2. Evrişim katmanı (Convolutional layer)

Evrişim katmanı CNN'in temel bir bileşenidir ve bir çıktı oluşturmak için iki girdiyi konvolüsyonlayarak çalışır. Dikey ve yatay kayan filtreler aracılığıyla tüm bölgeyi kapsayan giriş görüntüsünün özelliklerini kaydeder [25]. Evrişim Katmanı giriş verileri ve bir çekirdek alınarak gerçekleştirilen bir özellik haritası çıkarmak için giriş verilerine uygulanır. Görüntü girdileri için çekirdek, 0 ve 1'lerden oluşan 2 boyutlu bir dizidir ve elde edilen çıktı şekli çekirdek boyutlarına bağlıdır. Filtre, tüm genişliği tarayana kadar belirli bir Adım Değeri ile giriş görüntüsü boyunca yatay olarak hareket eder. Ardından, aynı Adım Değeri ile görüntünün başına dikey olarak geri atlar ve tüm görüntü taranana kadar işlemi tekrarlar. Bu çıktı bir sonraki aşama için girdi görevi görür [26]. Oluşturulan evrişim modeli 1, 2 veya 3 katman olacak şekilde tasarlanmıştır.

3.2.3. Havuz katmanı (Pooling layer)

Havuzlama katmanı, hesaplama yükünü azaltmak için ağ parametrelerini en aza indirmeyi amaçlar. Bu katman özellik haritalarının çözünürlüğünü azaltarak kayma varyansına ulaşmaya çalışır. Bu işlem görüntüde piksel kaybı oluşturabilirken bir sonraki katman için daha az hesap yükü oluşturur. Tipik olarak, iki evrişim katmanı arasına yerleştirilir ve bir havuzlama katmanının her bir özellik haritası, önceki evrişim katmanına

karşılık gelen özellik haritasına bağlanır [27]. Oluşturulan modelde havuz katmanı 1 ile 5 katman arasında olacak şekilde tasarlanmıştır.

3.2.4. Aktivasyon katmanı (Activation Layer)

Aktivasyon fonksiyonları, giriş verilerindeki doğrusal olmayan özelliklerin yakalanmasına yardımcı olarak uygulanan matematiksel işlemlerdir. Bu ağın daha karmaşık özellikleri öğrenmesini sağlar [28]. Bu çalışmada Sigmoid ve ReLU fonksiyonları uygulanmıştır. Şekil 4'te fonksiyonlara ait grafikler verilmiştir.

Sigmoid Fonksiyonu: Doymuş fonksiyonlardan biridir. Girdi aralığını 0 ile 1 arasına sıkıştırır. Çıktılarını olasılık değerleri olarak yorumlayabilmesi nedeniyle ikili sınıflandırma problemlerinde son katmanın aktivasyon fonksiyonu olarak yaygın kullanımı vardır [29]. Sigmoid fonksiyonu matematiksel olarak aşağıdaki gibi ifade edilir:

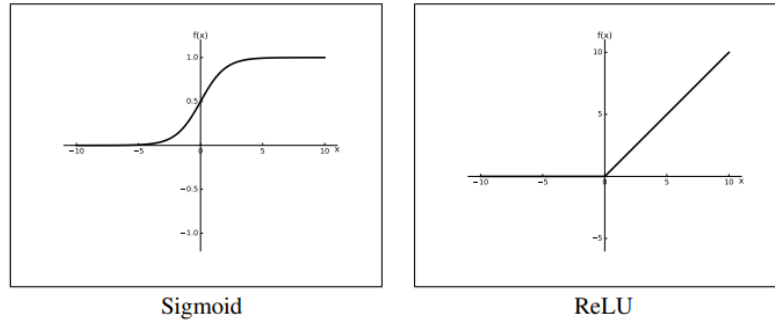
$$f(x) = 1/(1 + e^{-x}) \quad (1)$$

Bu fonksiyon, girdi değerleri arttıkça 1'e yaklaşır ve girdi değerleri negatifleştikçe 0'a yaklaşır.

ReLU Fonksiyonu: Giriş değeri pozitifse onu olduğu gibi bırakır, negatifse sıfıra eşitler [30]. Yani, ReLU fonksiyonunun matematiksel ifadesi şöyledir:

$$f(x) = \max(0, x) \quad (2)$$

ReLU, özellikle derin sinir ağlarında kullanıldığında eğitimi hızlandırabilir ve daha iyi sonuçlar elde edebilir. Bunun nedeni, ReLU'nun türevinin büyük çoğunlukla 1 veya 0 olmasıdır, bu da gradientleri daha iyi iletebilir hale getirir.



Şekil 4. Sigmoid ve ReLU Fonksiyon Grafikleri[34]
(Sigmoid and ReLU Function Graphs)

3.2.5. Tam bağlantılı katmanlar (Fully connected layer)

Bu katman genelde CNN'in son katmanları olarak kullanılmaktadır. Hedef çıktı sonucuna ulaşmak için gerekli olan belirli "bileşen" karışımını belirleyerek önceki katmanın özelliklerinin matematiksel bir toplamını gerçekleştirirler. Tam bağlantılı bir katmanda, her bir çıktı özellik öğesinin hesaplanmasında önceki katmanın tüm özellik öğeleri kullanılır [31].

3.2.6. Çıkış katmanı (Output layer)

İleri beslemeli bir sinir ağında, çıkış katmanı nöron birimleri sabit kalır ve önceki nöronlarla tam bağlantılar kurar. Tahmin yapmak için nihai eşik görevi görürler [27].

3.3. Hiper parametre optimizasyonu (Hyper parameter optimization)

Hiper parametre optimizasyonu (HPO), sınıflandırma ve tahmin doğruluğunu iyileştirmeyi, özellik seçim performansını artırmayı ve daha hassas çalışmalar için algoritmalar tarafından seçilen parametreleri daha uygun şekilde ayarlamayı amaçlayan bir değerler kümesidir. Hiper parametre optimizasyonu, belirli bir zaman diliminde hiper parametre değerlerinin optimum kombinasyonunu bulmayı amaçlayan bir işlemdir.

Bu hedeflere ulaşmak için en uygun test algoritmasını seçmek çok önemlidir [32]. HPO algoritması yalnızca ayrık, sıralı ve sürekli değişkenleri optimize etmekle kalmaz, aynı zamanda hangi değişkenlerin optimize edilmesi gerektiğini de belirler [33]. Izgara Arama, Rastgele Arama, Bayes yöntemi Genetik Algoritma ve Parçacık Sürü Optimizasyonu en çok kullanılan hiper parametre optimizasyon teknikleri olarak karşımıza çıkmaktadır.

Makine öğrenimi algoritmalarında hiper parametrelerin ince ayarı, ortaya çıkan modelin performansı üzerinde doğrudan etkisi olan çok önemli bir süreçtir. Hiper parametrelerin ayarları manuel olarak ayarlanabilir veya otomatik olarak optimize edilebilir.

- *Manuel hiper parametre ayarı*, hiper parametre değerlerinin uzman bilgisine dayalı olarak ve deneme yanılma yoluyla belirlenmesi yöntemidir. Bu yaklaşımda, deneyimli uzmanların bilgi ve deneyimleri, hiper parametrelerin optimal değerlerini etkili bir şekilde belirleyebilir. Ancak, manuel kurulum zaman alıcı ve hataya açık olabilir [34].
- *Otomatik hiper parametre ayarı*, hiper parametre değerlerini otomatik olarak belirlemek için makine öğrenimi algoritmalarında kullanılan bir yöntemdir. Bu yaklaşım genellikle önceden belirlenmiş bir arama stratejisi veya optimizasyon algoritması kullanır. Örneğin izgara arama, rastgele arama, Bayes optimizasyonu ve genetik algoritmalar gibi yöntemler kullanılabilir. Otomatik hiper parametre ayarı, hiper parametrelerin optimal değerini daha etkili ve verimli bir şekilde bulmaya yardımcı olabilir [35].

3.3.1. Açık kaynak hiper parametre optimizasyon çerçeveleri (Open source hyper parameter optimization frameworks)

HPO sorunlarının üstesinden gelmek için derin öğrenime geliştiricilerinin kullandığı için birçok açık kaynak hiper parametre kitaplığı mevcuttur [36]. Bu bölümde, çalışmada kullanılan açık kaynaklı HPO kütüphaneleri olan Optuna, HyperOpt ve Scikit-optimize yer verilmiştir.

Hyperopt , optimizasyon algoritmaları olarak RS ve BO-TPE içeren bir HPO çerçevesidir. Yalnızca tek bir modeli destekleyen diğer bazı kitaplıkların aksine, Hyperopt, hiyerarşik hiperparametreleri modellemek için birden çok model kullanabilir. Ek olarak Hyperopt, hiperparametre kombinasyonlarını depolamak için merkezi veritabanı olarak MongoDB kullandığından paralelleştirilebilir. Hyperopt-sklearn ve hyperas , Hyperopt'u scikit-learn ve Keras kitaplıklarına uygulayan iki kütüphanedir [36].

2019'da geliştirilen açık kaynaklı bir hiperparametre ayarlama optimizasyonu olan Optuna, hiperparametre optimizasyonunun uyum-deneme ve hata hesaplama sürecini otomatikleştirir [37]. Optuna, hedefli bir API yaklaşımı kullanarak en iyi performans için optimum hiperparametre değerlerini belirler.

Skopt olarak da bilinen Scikit-optimize, Bayesian optimizasyon ve sıralı model tabanlı optimizasyon yöntemlerini birleştiren esnek bir kütüphanedir. Hiperparametrelerin verimli bir şekilde ayarlanmasını sağlayan basit optimizasyon fonksiyonları sağlar [38].

4. Deneysel Sonuçlar (Experimental Results)

Bu çalışmada Mnist veri seti kullanılarak el yazısı rakamların derin öğrenme ile tespiti ve hiper parametre optimizasyonu kullanılarak öğrenme başarısının artırılması ve optimizasyon analizlerini ortaya koymak hedeflenmiştir. Hiper parametre optimizasyonu için açık kaynak kodlu Hyperopt, Optuna ve Scikit-Optimize kütüphaneleri kullanılmıştır. Rastgele parametrelerle model çalıştırılmış ve elde edilen sonuçlar Tablo 1' de verilmiştir.

Tablo 1. Rastgele parametrelerle elde edilen sonuçlar (Results obtained with random parameters)

Optimize edicinin öğrenme oranı	Dönüşüm katmanlarının sayısı	Her yoğun katmandaki düğüm sayısı	Her yoğun katmandaki düğüm sayısı	Tüm katmanlar için aktivasyon fonksiyonu	Başarı Yüzdesi%		
					Optuna	HyperOpt	Sci-Opt
1e-05	2	3	256	Relu	78.45	75.62	76.95
1e-03	3	4	150	Relu	97.13	97.46	97.27

El yazısı rakamların tanınması için geliştirilen modelin eğitimi ve analizler, Python programlama dili kullanılarak I5 11400h işlemci, 40 GB Ram ve GTX 1650 Ekran kartına sahip bir bilgisayarda gerçekleştirilmiştir. Geliştirilen CNN modeli için epoch sayısı 3 ve batch size değeri 128 olarak belirlenmiştir. Her hiper parametre tekniği için hiper parametre uzayı 10 kez çalıştırılmıştır. Çalışmada kullanılan hiper parametre uzayı Tablo 2' de verilmiştir.

Tablo 2. Hiper parametre uzayı (Hyperparameter space)

Optimize edicinin öğrenme oranı		Dönüşüm katmanlarının sayısı		Her yoğun katmandaki düğüm sayısı		Yoğun katmandaki düğüm sayısı		Tüm katmanlar için aktivasyon fonksiyonu	
EY	ED	EY	ED	EY	ED	EY	ED	Kategori	
0.000001	0.01	1	3	1	5	5	512	Relu	Sigmoid

ED: En düşük, EY: En yüksek

El yazısı rakamların tanınması için kullanılan Mnist veri setinin %25, %50, %75 ve %100'ü kullanılarak eğitim ve test verileri oluşturulmuştur. Model, veri setindeki toplam örneklem büyüklüğünün yüzde 10'u kullanılarak rastgele oluşturulan bir örnek kümesi ile test edilmiştir. Hyperopt, Optuna ve Scikit-Optimize kütüphaneleri kullanılarak derin öğrenme modeli optimize edilmiştir.

Optimizasyon 1. Veri Setinin %25'i kullanılarak el yazısı rakamların derin öğrenme ile tespiti

Optimizasyon sonuçlarında başarı oranı en yüksek sonuç HyperOpt kütüphanesi ile elde edilmiştir. Optuna kütüphanesinin eğitim sürecinin diğer kütüphanelere oranla daha kısa olduğu gözlemlenmiştir. Veri setinin yüzde 25'i kullanılarak elde edilen optimizasyon sonuçları Tablo 3.' de verilmiştir

Tablo 3. Veri Setinin %25'i için optimizasyon sonuçları (Optimization results for 25% of the Data Set)

	Optimizasyon süresi	Başarı Yüzdesi%
HyperOpt	415 saniye	98,80
Optuna	151 saniye	97,33
Scikit-Optimize	355 saniye	97,89

Optimizasyon 2. Veri Setinin %50'si kullanılarak el yazısı rakamların derin öğrenme ile tespiti

Optimizasyon sonuçlarında başarı oranı en yüksek sonuç Optuna kütüphanesi ile elde edilmiştir. HyperOpt kütüphanesinin eğitim sürecinin diğer kütüphanelere oranla daha kısa olduğu gözlemlenmiştir. Veri setinin yüzde 50'si kullanılarak elde edilen optimizasyon sonuçları Tablo 4.' de verilmiştir.

Tablo 4. Veri Setinin %50'si için optimizasyon sonuçları (Optimization results for 50% of the Data Set)

	Optimizasyon süresi	Başarı Yüzdesi%
HyperOpt	720 saniye	97,75
Optuna	280 saniye	98,06
Scikit-Optimize	798 saniye	97,46

Optimizasyon 3. Veri Setinin %75'i kullanılarak el yazısı rakamların derin öğrenme ile tespiti

Optimizasyon sonuçlarında başarı oranı en yüksek sonuç HyperOpt kütüphanesi ile elde edilmiştir. Optuna kütüphanesinin eğitim sürecinin diğer kütüphanelere oranla daha kısa olduğu gözlemlenmiştir. Veri setinin yüzde 75'i kullanılarak elde edilen optimizasyon sonuçları Tablo 5.' de verilmiştir.

Tablo 5. Veri Setinin %75'i için optimizasyon sonuçları (Optimization results for 75% of the Data Set)

	Optimizasyon süresi	Başarı Yüzdesi%
HyperOpt	1073 saniye	98,70
Optuna	392 saniye	98,61
Scikit-Optimize	1492 saniye	97,17

Optimizasyon 4. Veri Setinin %100'ü kullanılarak el yazısı rakamların derin öğrenme ile tespiti

Optimizasyon sonuçlarında başarı oranı en yüksek sonuç HyperOpt kütüphanesi ile elde edilmiştir. Optuna kütüphanesinin eğitim sürecinin diğer kütüphanelere oranla daha kısa olduğu gözlemlenmiştir. Veri setinin yüzde 100'ü kullanılarak elde edilen optimizasyon sonuçları Tablo 6.' de verilmiştir.

Tablo 6. Veri Setinin %100'ü için optimizasyon sonuçları (Optimization results for 100% of the Data Set)

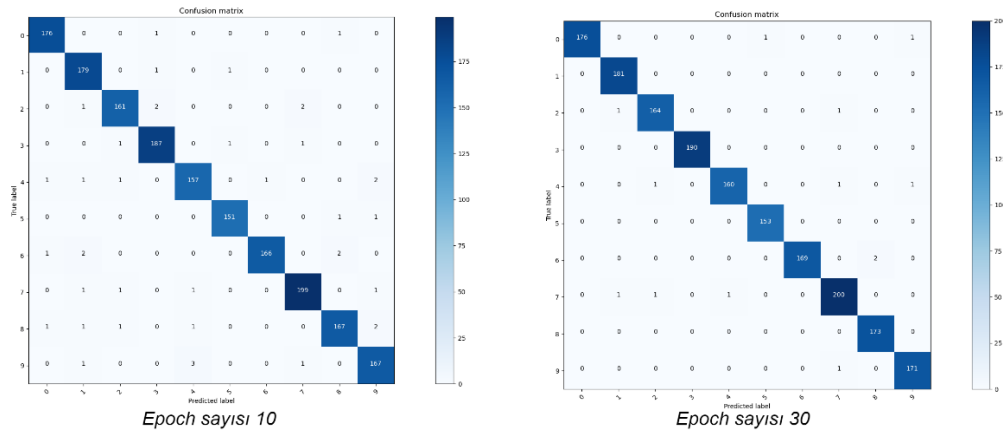
	Optimizasyon süresi	Başarı Yüzdesi%
HyperOpt	1756 saniye	98,87
Optuna	853 saniye	98,59
Scikit-Optimize	1547 saniye	98,49

Optimizasyon 5. Epoch(döngü) sayısının değişimi

Eğitim süreci boyunca tüm eğitim verilerinin ağı sunulma sayısına epoch sayısı denir. CNN modeli için epoch sayısı 10 ve 30 olarak belirlenmiş veri setinin %25'i kullanılmıştır. Diğer optimizasyon denemelerinde Optuna kütüphanesinin diğer kütüphanelere göre daha kısa zamanda hiper parametre optimizasyonu sağladığı için Optuna kütüphanesi kullanılmıştır. Epoch değerinin artırılması modelin süresini uzatırken başarı oranının da artmasını sağlamıştır. Elde edilen sonuçlar ve parametreler Tablo 7.' de verilmiştir. Ek olarak, analiz sonucunda elde edilen karmaşık matris tablosu Şekil 5'te gösterilmiştir.

Tablo 7. Epoch değeri 10 ve 30 için optimizasyon sonuçları (Optimization results for epoch value 10 and 30)

Epoch	Optimizasyon süresi	Parametre		Başarı Yüzdesi%
10	960 saniye	activation	relu	97,71
		learning_rate	0.009081356705666793	
		num_conv_layers	3	
		filters	32	
		Kernel_size	5	
		Strides	1	
		num_dense_layers	2	
		units	303	
		optimizer_name	RMSprop	
30	2160 saniye	activation	relu	99,26
		learning_rate	0.0035201802253737057	
		num_conv_layers	2	
		filters	16	
		Kernel_size	5	
		Strides	1	
		num_dense_layers	1	
		units	36	
		optimizer_name	Adam	



Şekil 5. Epoch değeri 10 ve 30 için karmaşıklık matrisi
(Confusion matrix for epoch value 10 and 30)

5. Sonuçlar ve Tartışma (Results and Discussion)

MNIST veri seti kullanılarak el yazısı rakamların evrişimli sinir ağı ile yüksek doğruluk ile tanınabilmektedir. Bu çalışmada, el yazısı rakamların tanınmasında açık kaynak kodlu hiper parametre optimizasyonu kütüphanelerinden Optuna, HyperOpt ve Scikit-Optimize kullanılmış, hiper parametrelerin ve epoch değerinin el yazısı tanımadaki başarısı değerlendirilmiştir. Kullanılan hiper parametre optimizasyon kütüphanelerinin farklı veri seti büyüklüklerinde optimizasyon süreleri ve başarıları analiz edilmiştir. Optimizasyon süresi incelendiğinde, Optuna kütüphanesi diğer hiper parametre optimizasyon kütüphanelerine göre daha kısa sürede tamamlamıştır. Başarı yüzdeleri birbirlerine yakın olsa da HyperOpt kütüphanesi diğer kütüphanelerden daha öndedir. Veri seti büyüklüğü ve epoch değerinin artması el yazısı rakamların tanınmasında başarı yüzdesini artırmıştır. Optimize edilmiş hiper parametrelerle eğitilen model, %99 doğruluk elde etmiştir. En iyi performansı elde etmek için yapılan çeşitli deneyler, belirli bir öğrenme hızı, dönem sayısı ve gizli katman düğümlerinin sayısı kombinasyonunun modelin genel performansını iyileştirdiğini göstermiştir. Ancak hiper parametre optimizasyon işleminin bilgisayar kaynakları ve zaman açısından maliyetli olduğunu belirtmek gerekir. Optimum hiper parametreleri belirlemek için daha büyük bir hiper parametre arama alanı veya daha fazla hesaplama gücü gerekebilir. Sonuç olarak bu çalışma, hiper parametre optimizasyonunun MNSIT veri seti üzerindeki el yazısı rakam tanıma çalışmalarında başarıyla uygulanabileceğini ve açık kaynak hiper parametre optimizasyon kütüphanesinin modelin performansını önemli ölçüde artırabileceğini göstermektedir. Hiper parametre analizi sonuçları, modelin performansının hiper parametre seçiminden önemli ölçüde etkilendiğini göstermektedir. Bu nedenle, el yazısı rakam tanıma için bir evrişimli sinir ağı kullanılırken, uygun hiper parametrelerin seçilmesi önemlidir. Elde edilen sonuçlar, doğru hiperparametre seçiminin modelin başarısını büyük ölçüde etkilediğini göstermektedir. Bu bulgular, benzer uygulamalarda model performansını artırmak için kullanılabilir ve ileriye yönelik araştırmalara ışık tutabilir.

Çıkar Çatışması Beyanı (Conflict of Interest Statement)

Yazarlar tarafından herhangi bir çıkar çatışması bildirilmemiştir.

Kaynaklar (References)

- [1] W. Ertel, *Introduction to Artificial Intelligence*, Springer Cham., 2017
- [2] A. Baldominos, Y. Saez, and P. Isasi, "A survey of handwritten character recognition with MNIST and EMNIST," *Applied Sciences*, vol. 9, no. 15. Aug. 01, 2019. doi: 10.3390/app9153169.
- [3] E. Kussul, T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database", *Image and Vision Computing*, vol.22, no.12, pp. 971-981, 2004, doi: 10.1016/j.imavis.2004.03.008.
- [4] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.
- [5] R. S. Humera and R. Zaheer, "Impact of Hyperparameters on Model Development in Deep Learning," in *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Springer, 2021, pp. 57–67.

- [6] R. Elshawi, M. Maher, and S. Sakr, "Automated Machine Learning: State-of-The-Art and Open Challenges," Jun. 2019, [Online]. Available: <http://arxiv.org/abs/1906.02287>. [Accessed: Dec. 28, 2023].
- [7] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization" in *Proceedings of the 24th International Conference on Neural Information Processing Systems NIPS 2011*. 2011, pp. 2546–2554.
- [8] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter, "SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization", *The Journal of Machine Learning Research*, vol. 23, no. 1, Jan. 2022.
- [9] L. Li, K. Jamieson, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," *Journal of Machine Learning Research*, vol.18, no.1, pp.6765–6816, Jan 2017, doi: 10.5555/3122009.3242042.
- [10] S. L. Chooi and A. S. Ghafar, "Handwritten Character Recognition Using Convolutional Neural Network", *Progress in Engineering Application and Technology*, vol. 2, no. 1, pp. 593–611, Jun. 2021.
- [11] O. M. Khanday, S. Dadvandipour, and M. A. Lone, "Effect of filter sizes on image classification in CNN: A case study on CFIR10 and fashion-MNIST datasets," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, pp. 872–878, Dec. 2021, doi: 10.11591/ijai.v10.i4.pp872-878.
- [12] L. Ming Seng, B. Bang Chen Chiang, Z. Arabee Abdul Salam, G. Yih Tan, and H. Tong Chai, "MNIST handwritten digit recognition with different CNN architectures," *Journal of Applied Technology and Innovation*. vol. 5, no.1,pp. 7-10, 2021.
- [13] H. Shao, E. Ma, M. Zhu, X. Deng, and S. Zhai, "MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization," *Intelligent Automation and Soft Computing*, vol. 36, no. 3, pp. 3595–3606, 2023, doi: 10.32604/iasc.2023.036323.
- [14] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 13, Dec. 2023, doi: 10.1002/widm.1484.
- [15] A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," *Neurocomputing*, vol. 283, pp. 38–52, 2018, doi: 10.1016/j.neucom.2017.12.049.
- [16] Y. LeCun, "The MNIST database of handwritten digits.", 1998. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>, [Accessed: Dec. 28, 2023].
- [17] M. Wu and Z. Zhang, "Handwritten Digit Classification using the MNIST Data Set Handwritten Digit Classification using the MNIST Data Set 1," 2014. [Online]. Available: <https://www.researchgate.net/publication/228685853> [Accessed: Dec. 28, 2023].
- [18] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *J Big Data* vol.8, no.53, pp. 1-74, Mar 2021. <https://doi.org/10.1186/s40537-021-00444-8>
- [19] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J Physiol*, vol. 195, no. 1, pp. 215–243, 1968, doi: [10.1113/jphysiol.1968.sp008455](https://doi.org/10.1113/jphysiol.1968.sp008455).
- [20] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology", *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 01, 2018. doi: 10.1007/s13244-018-0639-9.
- [21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Association for Computing Machinery*, vol.60, pp. 84-90, June 2017, doi: 10.1145/3065386
- [23] M. Krichen, "Convolutional Neural Networks: A Survey," *Computers*, vol. 12, no. 8, 2023, doi: 10.3390/computers12080151.
- [24] Ö. İnik Ve E. Ülker, "Derin Öğrenme Ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri", *Gbad*, Vol. 6, no. 3, pp. 85–104, 2017.
- [25] S. Ali, Z. Shaukat, M. Azeem, Z. Sakhawat, T. Mahmood, and K. ur Rehman, "An efficient and improved scheme for handwritten digit recognition based on convolutional neural network," *SN Appl Sci*, vol. 1, no. 9, Sep. 2019, doi: 10.1007/s42452-019-1161-5.
- [26] M. Sahu and R. Dash, "A Survey on Deep Learning: Convolution Neural Network (CNN)," 2021, *Smart Innovation, Systems and Technologies*, vol.153, pp. 317–325, 2021. doi: 10.1007/978-981-15-6202-0_32.
- [27] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit*, vol. 77, pp. 354–377, 2018, doi: [10.1016/j.patcog.2017.10.013](https://doi.org/10.1016/j.patcog.2017.10.013).
- [28] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric Defect Detection Using Activation Layer Embedded Convolutional Neural Network," *IEEE Access*, vol. 7, pp.70130-70140, Apr. 2019, doi: 10.1109/ACCESS.2019.2913620.

- [29] M. Tanaka, "Weighted sigmoid gate unit for an activation function of deep neural network," *Pattern Recognit Lett*, vol. 135, pp. 354–359, 2020, doi: 10.1016/j.patrec.2020.05.017.
- [30] M. Coşkun, A. Uçar, Ö. Yildirim, and Y. Demir, "Face recognition based on convolutional neural network," in *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, 2017, pp. 376–379. doi: 10.1109/MEES.2017.8248937.
- [31] S. L. Hijazi, R. Kumar, and C. Rowen, "Using Convolutional Neural Networks for Image Recognition," *Cadence Design Systems Inc*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6212567>
- [32] F. Yılmaz and M. C. Kasapbaşı, "Eeg Sinyalleri İle Epilepsi Krizinin Tahminlenmesinde Rassal Orman Algoritması İle Hiper Parametre Optimizasyonun Uygulanması," *İstanbul Ticaret Üniversitesi Teknoloji Ve Uygulamalı Bilimler Dergisi*, vol. 3, no. 2, pp. 189–203, 2021.
- [33] T. Bartz-Beielstein, "Hyperparameter Tuning and Optimization Applications," in *Hyperparameter Tuning for Machine and Deep Learning with R*, Springer, 2023, pp. 165–175.
- [34] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," *Journal of Machine Learning Research*, vol.13, no.10, pp.281-305. Feb 2012.
- [35] F. Hutter, L. Kotthoff, and J. Vanschoren, "Automated Machine Learning: Methods, Systems, Challenges," *Automated Machine Learning*, pp. 113–134. Springer. 2019.
- [36] L. Yang And A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol.415, pp.295-316,November 2020. <https://doi.org/10.1016/j.neucom.2020.07.061>
- [37] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-Generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in KDD '19*. New York, NY, 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.
- [38] "scikit-optimize: Sequential model-based optimization in Python — scikit-optimize 0.7.3 documentation. (n.d.)." [Online]. Available: <https://scikit-optimize.github.io/>. [Accessed: Dec. 28, 2023].

* This paper was presented at the 5th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAAME 2023) and the abstract was published as an e-book.

This is an open access article under the CC-BY license

