YAKINLIK SENSÖR DİZİLERİNDE YAPAY ZEKA YONGASI TABANLI HAREKET ALGILAMA VE TANIMA

ARTIFICIAL INTELLIGENCE CHIP BASED MOTION DETECTION AND RECOGNITION IN PROXIMITY SENSOR ARRAYS

Mehmet Ali KANDİLCİK

Doç. Dr. Dinçer GÖKCEN

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Elektrik-Elektronik Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2025

ÖZET

Yakınlık Sensör Dizilerinde Yapay Zeka Yongası Tabanlı Hareket Algılama ve Tanıma

Mehmet Ali KANDİLCİK

Yüksek Lisans, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Dinçer Gökcen

Temmuz 2025, 85 sayfa

Bu tez çalışmasında kapasitif yakınlık sensör dizilerinden toplanan verilerin eğitilmiş ağırlık ve sapma değerleri kullanılarak, el hareketlerini tanıma yeteneğine sahip bir yapay zeka yongası tasarımının aşamaları sunulmuştur. Literatürde, hareket algılama ve tanıma için birçok farklı sistem bulunmaktadır. Bu çalışmada hareket algılama ve tanıma için birçok yazılım ve donanım isterinin tek bir tümleşik devre ile yani yapay zeka yongası ile gerçekleştirilmesi amaçlanmıştır. Tez çalışmasında, kapasitif yakınlık sensör dizileri kullanılarak toplanan veri seti kullanılmıştır. El hareketlerinden alınan verilerin %75'i eğitim, %15'i doğrulama ve %10'u test verisi olacak şekilde ayrılmıştır. Hareket algılama ve tanıma için yapay zeka yongası girdi ve çıktı katmanları dahil 4 katmandan oluşan MLP (İng. Multi Layer Perceptron) mimarisini kullanmaktadır. Her katmanda sırasıyla: 36, 20, 20 ve 4 düğüm bulunmaktadır. MLP mimarisi, bilgisayar ile benzetim ortamında %99,58 doğruluk ve %99,50 F1 puanına ulaşabilmiştir. Çalışmadan elde edilen analiz sonuçlarına göre sayısal tasarımda gerçekleştirilecek mimarinin temel yapı taşları belirlemiştir. Bilgisayar ortamında çalışan algoritma temel alınarak sayısal tasarımı yapılmış ve son aşama olan uygulamaya özgü yapay zeka yongası tasarlanmıştır. Sayısal tasarım, 16 bit sabit nokta sayı formatını kullanmaktadır. Uygulamaya özgü tümleşik devre (ASIC) konseptiyle RTL2GDSII yaklaşımı kullanılarak tasarımı yapılan yapay zeka yongası, ticari FPGA SoC kullanılarak yapılan tasarımdan 4 kat daha yüksek frekansa ulaşabilirken, 7,25 kat daha düşük güç tüketmektedir. Tez çalışması kapsamında, AI (İng. Artificial Intelligence) yonga konseptinde bir tasarım gerçekleşmiş ve belirlenen bir problem için yapay zeka yongasını içeren sistemin ve alt-sistemlerin tasarım akış süreci üzerine bir yöntem sunulmuştur.

Anahtar Kelimeler: Yapay zeka yongası, yakınlık sensör dizileri, yapay sinir ağları, sayısal tasarım, ASIC

ABSTRACT

Artificial Intelligence Chip Based Motion Detection and Recognition in Proximity Sensor Arrays

Mehmet Ali KANDİLCİK

Master's Degree, Department of Electrical and Electronics Engineering

Supervisor: Doç. Dr. Dinçer Gökcen

July 2025, 85 pages

In this thesis, the stages of designing an artificial intelligence chip capable of recognizing hand gestures are presented, utilizing the trained weight and bias values derived from data generated by capacitive proximity sensor (CPS) arrays. Numerous systems for motion detection and recognition exist in the literature. The study aims to fulfill various software and hardware requirements for motion detection and recognition using a single integrated circuit, specifically an artificial intelligence (AI) chip. A dataset collected using CPS arrays was utilized in the thesis study. 75% of the data obtained from hand gestures was set aside as training, 15% as validation, and 10% as test data. For motion detection and recognition, an AI chip was designed to run a Multi-Layer Perceptron (MLP) architecture, which consists of 4 layers, including input and output layers. Each layer has 36, 20, 20 and 4 nodes, respectively. MLP architecture achieved 99.58% training accuracy and 99.50% F1 score in a computer-based simulation environment. According to the analysis results obtained from the study, the basic building blocks of the architecture to be implemented in the digital design were determined. The digital design was created based on the algorithm tested in the computer environment, and at the final stage, an application-specific AI chip was designed. The digital design used a 16-bit fixed-point number format. AI chip, designed using the RTL2GDSII approach with the application-specific integrated circuit (ASIC) concept, achieved a 4 times higher frequency than the design made using a commercial FPGA SoC while consuming 7.25 times less power. Within the scope of the thesis study, a design was realized for the AI chip concept, and a method was presented for the design flow process of the system and subsystems containing the artificial intelligence chip for a specific problem.

Keywords: Artificial intelligence chip, capacitive proximity sensors, artificial neural network, digital design, ASIC

TEŞEKKÜR

Yüksek lisans öğrenim sürecim boyunca, kapsamlı ve çok yönlü literatür bilgisi sayesinde tez konusunun belirlenmesinde ve tez çalışması süresince de birçok alanda sahip olduğu tecrübeleriyle beraber derin bilgi birikimi sayesinde adeta akıl hocalığı yapan ve tez çalışmasının her aşamasında en uygun ve en doğru yöntem ile bana rehberlik eden danışmanım Sayın Doç. Dr. Dinçer Gökcen'e,

Tez jürimde yer alan ve bilgi birikimlerini benimle paylaşan değerli hocalarıma,

Tez çalışması süresince hem teknik hem de yöntem bakımından sağlamış olduğu katkılardan dolayı Dr. Kübra Saka'ya,

Tez çalışmalarında vermiş oldukları büyük katkılardan dolayı Hacettepe Üniversitesi Elektrik-Elektronik Mühendisliği Mikro ve Nano Sistemler Araştırma Laboratuvarı (MNS-Lab.) öğrencilerine, özellikle sensör tasarımında ve veri setinin oluşturulmasında önemli katkılarından dolayı Hazar Haluk Aksaç başta olmak üzere, Sena Oğuz ve Turgut Toprakcı'ya,

Tez çalışmalarımda desteklerini esirgemeyen Aselsan'a, Cadence Üniversite Programı'na (İng. Cadence University Program),

Doğduğum zamandan itibaren beni karşılıksız biçimde destekleyen ve hayatım boyunca her zaman yanımda olan değerli annem ve babama,

İsmini sayamadığım ama tez çalışması süresine hem teknik hem de manevi olarak desteklerini esirgemeyen bütün değerli arkadaşlarıma,

Teşekkürlerimi sunarım.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

KISALTMALAR

Kısaltmalar

LLM	Büyük Dil Modelleri (İng. Large Language Model)
ML	Makine Öğrenmesi (İng. Machine Learning)
DL	Derin Öğrenme (İng. Deep Learning)
AI	Yapay Zeka (İng. Artificial Intelligence)
ANN	Yapay Sinir Ağları (İng Artificial Neural Network)
GPU	Grafik İşlem Birimi (İng. Graphics Processing Unit)
FPGA	Alanda Programlanabilir Kapı Dizisi (İng. Field Programmable Gate Array)
ASIC	Uygulamaya Özgün Tümleşik Devre (İng. Application Specific Integrated Circuit)
RTL	Yazmaç Transfer Seviyesi (İng. Register Transfer Level)
HDL	Donanım Tanımlama Dili (İng. Hardware Description Language)
CPS	Kapasitif Yakınlık Sensörü (İng. Capacitive Proximity Sensor)
MLP	Çok Katmanlı Algılayıcı (İng. Multi Layer Perceptron)
RTL2GDSII	Yazmaç Transfer Seviyesinden Grafik Tasarım Sistemine (İng. Register Transfer Level to Graphic Database System II)
CNN	Konvolüsyonel Sinir Ağları (İng. Convolutional Neural Network)
EDA	Elektronik Tasarım Otomasyonları (İng. Electronic Design Automation)
CLB	Yapılandırılabilir Mantıksal Bloklar (İng. Configurable Logic Blocks)
LUT	Başvuru Çizelgesi (İng. Lookup Table)
PCA	Temel Bileşen Analizi (İng. Principal Component Analysis)
ROM	Sadece Okuma Belleği (İng. Read Only Memory)

RAM	Rastgele Erişimli Bellek (İng. Random Access Memory)
FSM	Sonlu Durum Makinesi (İng. Finite State Machine)
DRC	Tasarım Kural Denetimi (İng. Design Rule Check)
CTS	Saat Ağacı Sentezi (İng. Clock Tree Synthesis)

1. GİRİŞ

Geçmişten günümüze teknolojik buluşların ve icatların temel konusu insan olmuştur. İnsanoğlu karşılaştığı problemlere kalıcı çözümler bulmak için teknolojik gelişmelere ve yeni çalışmalara devam etmektedir. 21. Yüzyılın önemli çalışma konularından birisi de yapay zekadır [1]. Yapay zeka hem gündelik hayatı hem de bilim camiasını oldukça yakında ilgilendiren disiplinler arası bir konulardan biridir. Gündelik hayatta büyük dil modelleri (İng. Large Language Model-LLM) başta olmak üzere 7'den 70'e yapay zeka kullanımının giderek arttığı görülmektedir. Bunun paralelinde bilim camiasında 2024 Nobel Fizik ödülünün, yapay zeka sistemleri ve tez konusunun ana arterlerinden biri olan yapay sinir ağları ile ilgili çalışmalarından dolayı John Hopfield ve Geoffrey Hinton'a verilmesi, yapay zekanın önemini göstermektedir.

Yapay zeka üzerine çalışmalara devam edildikçe, yapay zekanın kapsamında kavramsal yaklaşımlar bilim camiası tarafından türetilmiştir. Bunlardan makine öğrenmesi (İng. Machine Learning-ML), çeşitli algoritmalar ve teknikleri kullanarak kendisine yüklenen veriler üzerinden anlam ve cevap oluşturmaya çalışır [2]. Derin öğrenme (İng. Deep Learning-DL), beynin çalışmasını taklit etmek için, katmanlar içerisinde bulunan birçok düğümün birbiriyle etkileşimi sonucu oluşan bilgiyi analiz edip anlamlandırmayı amaçlar. İkisi arasındaki temel fark ML, kendisine sunulan verileri işlerken kullanıcının belirlediği parametreler doğrultusunda öznitelik çıkarımı yaparken DL, içerdiği çok katmanlı yapay sinir ağları (İng. Artificial Neural Networks-ANN) doğrultusunda kendi öznitelik çıkarımını gerçekleştirebilir [3]. Birçok ANN'den oluşan sistemler çok katmanlı algılayıcı (İng. Multi Layer Perceptron-MLP) olarak isimlendirilir [4]. MLP ile insanların hareketlerini yorumlayabilen, anlamlandırabilen ve çözümleyebilen sistemler tasarlanabilir. MLP mimarisi problemlere getirdiği çözümlerle, tezde kullanılan mimarilerden biri olmuştur.

Yapay zekanın hızla gelişmesini destekleyen faktörlerden biri de yazılımların çok sayıda fonksiyonu, çok daha kısa tasarım sürelerinde gerçekleştirebilme yeteneği olmuştur [1]. Bu tarz yazılım dilleri, yüksek seviye (İng. high level) diller olarak adlandırılır [5]. Bu dillere örneklerden biri de Python'dır. Python, sahip olduğu açık kaynak ve yüksek sayıdaki kütüphane içeriği ile çoğu yapay zeka uygulamasında da sıklıkla tercih edilen dillerden biridir [5]. Fakat, düşük seviyeli dillere göre daha yüksek donanım kapasite ihtiyacı ve algoritmanın daha yavaş çalışması gibi dezavantajlara da sahiptir [6].

İnsan beyninde bulunan nöronlar, haberleşmek için kullandığı çok sayıdaki sinaps bağlantısı sayesinde paralel işlem yapabilmektedirler. Sinapslar sayesinde beyin, sıralı işlemlere kıyasla aynı zamanda daha yüksek sayıda işlenmiş bilgi üretebilirler [7]. Tez kapsamında beyin anatomisi ilham alınarak probleme özgü mimari ve paralel işlem kabiliyetine sahip MLP yonga tasarımının gerçekleştirilmesi amaçlanmıştır.

İşlemci tabanlı mimariler genelde sıralı işlemler ile çıktı üretirler. Bu mimari MLP tasarımlarını gerçekleştirmede dezavantajlı olduğu için yapay zeka uygulamalarında bazı özelleşmiş donanımlar kullanılır. Bu donanımlar, yoğun seviyede paralel ve eş zamanlı matematiksel işlem yapabilirler [8]. Bunlar arasında grafiksel işlem birimi (İng. Graphics Processing Unit-GPU), alanda programlanabilir kapı dizisi (İng. Field-Programmable Gate Array-FPGA) ve uygulamaya özgü tümleşik devre (İng. Application Specific Integrated Circuit-ASIC) yer almaktadır [8]. Tez konusunda belirtilen problemi çözmek için probleme özgün tasarım yaklaşımları gerekmektedir. Bundan dolayı tez kapsamında donanım tasarımı seviyesinde sağladığı esneklikten dolayı FPGA ve ASIC tercih edilmiştir. FPGA ve ASIC, donanım tanımlama dili (İng. Hardware Description Language-HDL) ile donanım seviyesinde tasarıma imkan tanımaktadır.

FPGA, ASIC'ye göre daha düşük tasarım zamanı gerektiren sistemler olup, tekrar konfigüre edilme avantajı ve daha düşük maliyetler ile donanımı doğrulama avantajı sağlamasıyla tercih edilmektedir. Fakat ASIC, FPGA'ye göre doğrudan devre seviyesinde tasarım imkanları sağladığı için benzer mimaride daha düşük güç tüketimi ve daha yüksek hızlara ulaşabilir [8]. Karşılaştırmalar sonucunda, tez konusunda yapay zekanın etkin, verimli ve problemleri çözme rolündeki katalizör etkisini ivmelendirmesi amacıyla paralel işlem kabiliyetine sahip MLP donanım altyapısı, ASIC olarak tasarlanmıştır. Belirtilen özgün işlemler için tasarlanan donanım, fiziksel bir yonga serimine sahip olduğu için yapay zeka yongası olarak adlandırılmıştır [9].

Sensörler, teknolojik aygıtların duyu görevini üstlenirler [10]. Yapay zeka uygulamalarında, objeleri algılamak ve tanımak için kapasitif yakınlık sensörü (İng. Capacitive Proximity Sensors-CPS) gibi çeşitli sensörler kullanılmaktadır. CPS, muadillerine göre benzer boyutlarda daha uzun mesafe ölçüm alabilmesi, daha düşük maliyetler ile ölçülebilmesi ve daha farklı tasarımları (daire, kare, esnek vb.) desteklemesiyle yakınlık sensörü ile ilgili yayınlarda 2015'ten itibaren bilimsel çalışmalarda kullanım sıklığı artarak devam etmektedir [11].

CPS, kullanım esnekliği sayesinde obje algılamaya ihtiyaç duyulan uygulamalarda kullanılan sensör çeşitlerinden biri olmuştur [12]. Tez kapsamında sensör dizisi veya algılayıcı düzlemi, CPS'den oluşmuştur. El hareketinin konumuna göre, sensör dizisine ait her bir elektrottan analog işaret üretilir. Sonrasında bu analog işaret, sayısal işarete dönüştürülür. Sayısal işaretler ise, tasarlanmış MLP mimarisini içeren tümleşik devreye sürülür. Tasarlanmış MLP ise önceden eğitilmiş modelin ağırlık ve sapma değerlerini içerdiği için, uygulamaya özgü çözümler üretir. Böylelikle CPS ile uyumlu çalışabilecek yapay zeka yongası tasarlanabilir.

Yapay zeka uygulamalarında CPS ile MLP kullanımı, bilimsel camiada da tercih edilen yöntemlerden biridir [13]. Tez çalışması, uygulamaya özgü tasarımların, performans ve güç tüketimi konularında sağlayacağı avantajları değerlendirmeyi amaçlamıştır. Bu bağlamda öncellikle Python dilinde tasarlanan AI algoritması ile girdi verilerini oluşturan el model hareketleri koşturulmuştur. Buradan elde edilen ağırlık ve sapma değerleri kaydedilmiştir. Sonrasında oluşturulan MLP mimarisi, yazmaç transfer seviyesi (İng. Register Transfer Level-RTL) seviyesinde de tasarlanmıştır. En sonunda bu tasarım FPGA seviyesinde doğrulandıktan sonra, ASIC tasarımı yapılarak yapay zeka yongası içeren sistem tasarlanmıştır. Böylelikle gündelik hayattaki problemlere çözümler sağlaması amacıyla büyük boyutlu ve yüksek güç gibi isterlerden arındırılmış (GPU, CPU veya verimi daha düşük yüksek seviye diller) donanımlar yerine daha özel isterleri yerine getirmeye yoğunlaşmış bir tasarım tercih edilebilir.

1.1. Tezin Kapsamı

Ana konu olarak, yapay zeka yongası tasarımını ele alan tez, 2. Bölümde Genel Bilgiler başlığı altında bu ana konu çerçevesinde farklı sensör tiplerinin kullanımını, donanım gerçeklemesi için kullanılan cihazların birbirleri arasındaki avantajlarının ve dezavantajlarının karşılaştırılmasını ve literatür araştırmalarını içerir. 3. Bölümde Deneysel Çalışmalar başlığı altında tercih edilen tez tasarım sürecinin, yüksek seviye dillerle oluşturulan tasarımdan, düşük seviyeli dillere aktarımına kadar ilerlemenin detaylarından bahsedilmiştir. Detaylarda donanım altyapısı ve tasarlanan donanımın doğrulaması hakkında da bilgiler verilmiştir. 4. Bölümde, Oluşturulan Veri Seti Üzerinden Çalışmalar başlığı altında karşılaştırılmalı uygulama sonuçları, oluşturulan farklı veri setlerin birbirleri arasındaki avantajları ve dezavantajlarına bağlı olarak oluşabilecek farklı sonuçların beklentileri ne kadar karşıladığı ve beklentiler sonucunda oluşan tartışmalara yer verilmiştir. Bu başlık altında, RTL seviyesi tasarlanan MLP

mimarisinin FPGA üzerinde doğrulanıp, gerçekleştirilme akışından bahsedilmiştir. Elde edilen sonuçlar, sonradan ASIC ile karşılaştırılmak üzere ilgili başlık altında bahsedilmiştir. Son bölüm olan 5. Bölüm'de Yapay Zeka Yongası Tasarımı başlığı altında yapay zeka yongasının, yazmaç transfer seviyesinden grafik tasarım sistemine (İng. Register Transfer Level to Graphic Design System II-RTL2GDSII) aktarım sürecinden bahsedilmektedir. Bu başlık altında yapay zeka yongasının hem FPGA hem de ASIC'de tasarlanarak elde edilen sonuçlar karşılaştırılmaktadır. 6. Bölümde, Sonuçlar başlığı altında bütün tez çalışma sürecinin nihai sonuçları yer almaktadır. 7. Bölüm olan Yorumlar ve Gelecek Çalışmalar başlığı altında bütün çalışmalar gözetilerek, hangi süreçlerin ne kadar iyileştirilebileceği, tez çalışmasının geliştirilmesi gereken yönleri ve karşılaşılan bazı problemlerden bahsedilmektedir.

2. GENEL BİLGİLER

2.1. Kapasitif Yakınlık Sensörü

2.1.1 Yakınlık Algılama için Sensör Çeşitleri

Günümüzde hareket tanıma ve algılama için kullanıma bağlı farklı sensör çeşitleri kullanılmaktadır. Geleneksel seramik/metalik sensörlerin kırılgan materyallerinden kaynaklı kısıtları, üretim maliyetleri ve özellikle de boyutlarından ötürü, yakınlık algılamada birçok sensör çeşidi geliştirilmiştir [14]. Algılama tekniklerinden sıklıkla kullanılanlardan optik sistemler ışık kaynağını algılamak için iletken veya yalıtkan malzemelerden oluşturularak kullanılırlar. İndüktif tabanlı algılama yöntemleri ise sadece iletken objeleri algılamak için kullanılırlar. Kapasitif tabanlı algılama yöntemleri ise iletken ve yalıtkan malzemeleri algılamak amacıyla ihtiyaç isterlerine bağlı olarak üretilerek kullanılırlar [14]. Kapasitif tabanlı algılama yöntemleri daha esnek kullanım alanları, üretim ve uygulama kolaylığından dolayı sektörde sıklıkla tercih edilmektedirler [15]. Özetle, indüktif sensörlerin aksine, kapasitif yakınlık sensörler çevresindeki objelerin iletken ve farklı dielektrik özelliklerine bağlı olmaksızın daha kapsamlı tespit yeteneklerine sahiptirler. Farklı algılama yöntemleri, temel bileşenleri ve tespit edilen objelerle ilgili bilgilendirme Çizelge 2.1'de verilmiştir.

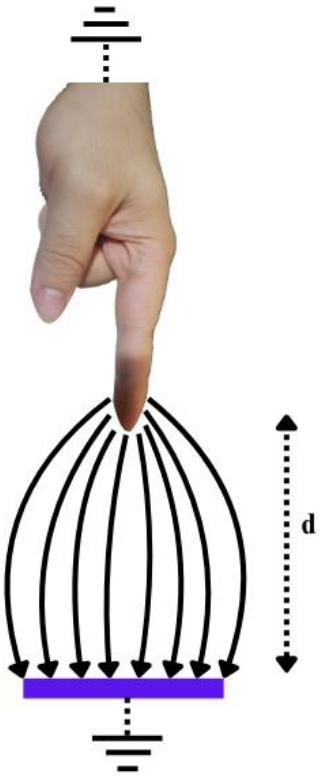
Çizelge 2.1. Yakınlık tespitinde kullanılan algılayıcıların yöntemleri [11].

Algılama Yöntemi	**Tespit Edilebilen Objeler**	**Algılayıcı Bileşeni**
Optik	İletken veya Yalıtkan	Işık Kaynağı
Ultrasonik	İletken veya Yalıtkan	İşaret Üreteci
İndüktif	Sadece İletken	İletken Sarmal
Kapasitif	İletken veya Yalıtkan	İletken Elektrot

2.1.2 Kapasitif Yakınlık Sensörüne Giriş

Kapasitif yakınlık sensörü teknolojisi, etkileşime girdiği objeden kaynaklı kapasitans değerinin değişmesini kullanarak, algılama sağlayan bir teknolojidir. Böylece yakınlık sensörü olarak özellikle, şekil tespiti, materyal sınıflandırması ve dokunsal algılama gibi

çeşitli alanlarda tercih edilirler [11, 12, 14]. Geleneksel kapasitörler çoğunlukla iki iletken katman arasına yerleştirilmiş bir yalıtkan malzemeden oluşmaktadır. CPS ise tek bir iletken katman üzerinden belirli uzaklıklardaki cisimlerle etkileşim kurabilmektedir. İletken katmana nesne yaklaştırıldıkça oluşan etkileşimle kapasitans değerinde değişimler gözlemlenir. Kapasitans değerini değişimi, uygun sinyal işleme yöntemlerinin uygulanması ile anlamlandırılır ve algılama sağlanır.



Şekil 2.1. Kapasitif yakınlık sensörün el ile etkileşimi.

Şekil 2.1'de kapasitif sensör elektroduna insan eli yakınlaştırılmasıyla kapasite değerinin değişimi sembolik olarak gösterilmiştir. Elektrot yüzeyinin alanının artışıyla algılama hassasiyeti arttırılabilir [16]. Fakat yüzey alanındaki artış ve farklı geometriler ortam gürültüsünden etkilenmesine veya istenmeyen nesnelerin algılanmasına neden olabilir. Tersi durum elektrot yüzeyini küçültüp, birden fazla elektrot kullanma yani kapasitif sensör dizisi kullanma için geçerlidir. Bu ikilemlerin avantajları ve dezavantajları değerlendirilerek, yakınlık algılama için kapasitif sensör dizisi tasarlanabilir.

Birden fazla CPS kullanılarak yapılan yakınlık algılamalarında algılanacak objenin daha fazla detay bilgisine ulaşılabilir [16]. Bu durumda dikkat edilmesi gereken nokta karşılıklı

kapasitans oluşturma durumudur. Çünkü her ne kadar objeyi detaylarıyla algılamak için CPS dizisi kullanılsa da bu CPS dizilerinin aralarındaki mesafenin yanlış belirlenmesi yakınlık algılamada sorunlar oluşturabilir [17]. Bunun sonucunda MLP, parazitlenmelerden dolayı gürültüyü algılayabilir hatta objeyi yanlış yorumlayabilir. Böylelikle eğitilen modelin doğruluk değeri düşük olur. Bunun için model tespiti sırasında: El hareketinin sabit olması, belirlenen her bir CPS elektrotunun boyutuna bağlı olarak aralarındaki mesafenin uygun tercih edilmesi gerekir [17]. MLP'yi parazitik etkilerden arındırılmış ve CPS dizisi üzerinde uygun biçimde konumlanmış el hareketleri gibi ideale yakın oluşturmak gerekir. Bunun için literatürde bulunan CPS elektrot konumlandırmanın özgün çalışmaya uygun biçimde yapılması gerekir [16, 17].

2.1.3 Literatürde Kapasitif Sensörlerin Kullanımı

Kapasitif sensör dizileri tasarımı isterlere bağlı olarak daire veya kare biçiminde tasarlanabilir. Hatta daha yüksek doğruluk seviyelerine ulaşılmak istenilirse dokunma algılayan kapasitif sensör dizileri de tasarlanabilmektedir. Böylece algılanacak objenin daha fazla detay bilgisine erişilebilir. Arimatsu ve Mori'nin çalışmasında [18], VR teknolojisi için kullanıcının el hareketi bilgilerini yüksek doğruluk ile işlemeleri gerektiği görülmektedir. Bu amaçla 62 CPS'i, bir oyun koluna monte etmişlerdir. El hareketleri ve el boyutları çok çeşitli parametreler içerdikleri için el hareketi ve el tipini tespit etmek için makine öğrenmesi yöntemini kullanmışlardır. Çalışmanın sonucunda sundukları değerlere göre, yüksek doğrulukta 3 boyutlu hareketli el görseli tespiti sağlayabilmişlerdir. Fakat yüksek donanım kapasitesi, sadece dokunma ile algılama ve yüksek seviyede dil ile eğitim modeli tasarımı, bazı kısıtların ortaya çıkmasına neden olmaktadır. Yüksek donanım isterine örnek olarak, gerçek zamanlı 90 FPS çalışma için Intel Core i7-6950X 3.0GHz CPU ve NVIDIA GTX1080 GPU tercih edilmiştir. Çalışma incelendiğinde yüksek güç tüketimi ve yüksek maliyetli donanımlar kullanılmıştır. Ek olarak çıkarılacak diğer sonuç ise kapasitif yakınlık sensörleri, yapısından dolayı pek çok farklı tasarıma monte edilip kullanılabilirler.

Literatürde CPS'nin kullanıldığı farklı uygulamalar bulunmaktadır. Fakat bu uygulamalarda, MLP mimarisini koşturan tümleşik devre tasarımı yerine, bilgisayar destekli uygulamalar üzerinden tahmin gerçekleştirilmiştir [19]. Bu yöntemi kullanan araştırmaların avantajları, HDL kullanarak RTL tasarlayan çalışmalara kıyasla daha farklı algoritmalar koşturup, bunları birbirleriyle kıyaslayabilmişlerdir. Böylece hangi algoritmanın, hangi özgün yöntem için daha uygun olacağına dair çeşitli çıkarımlar

yapabilmektedirler. Literatürde RTL seviyesinde gerçekleştirilen MLP tasarımları kullanılarak model tespiti yapan çalışmalar da bulunmaktadır. Fakat bu çalışmaların genelinde kamera veya çok yüksek sayıda sensör içeren sistemler kullanılmaktadır [20]. Literatür çalışmalarının incelenmesi sonucunda, CPS dizisi, muadillerine kıyasla düşük maliyetli ve kolay kullanımından dolayı tez çalışmasında, model veri seti oluşturmak için kullanılmıştır. Oluşturulan veri seti de kullanılarak AI modeli eğitilmiştir. Eğitim sonucu elde edilen ağırlık ve sapma değerleri RTL seviyesinde MLP mimarisinde saklanarak, model algılama ve tanıma gerçekleştirilmiştir.

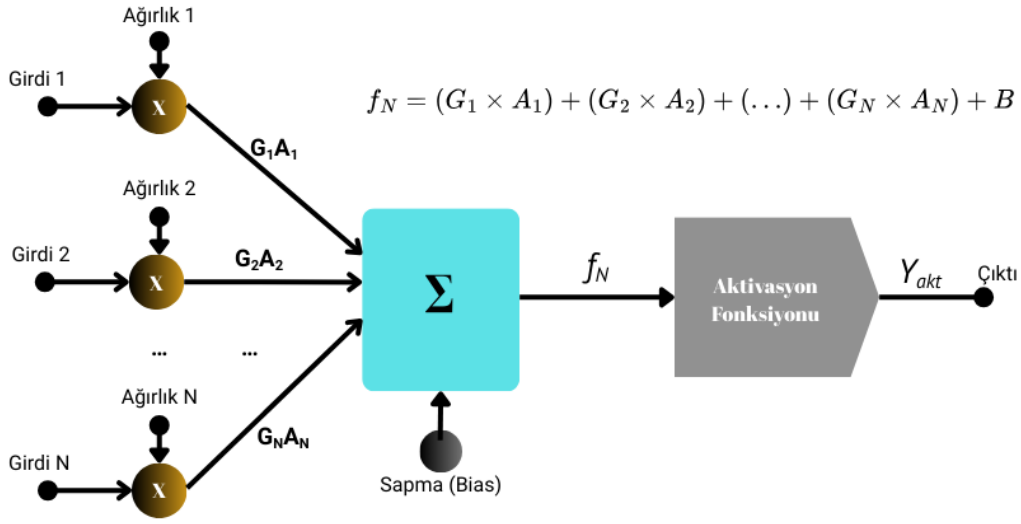
2.2. Yapay Sinir Ağları

2.2.1 Literatürde Model Tanıma ve Tespiti için Yapay Zeka Uygulaması

Yapay zekanın sistem seviyesinde kullanımının yaygınlaşmasının nedenlerinden birisi problemleri çözmek için akıllı sistemlere ihtiyaç duyulmasıdır [21]. Bu hedef doğrultusunda yapay zeka, sürekli yöntem iyileştirmeleri ve güncellemelerle farklı teknolojiye sahip cihazlarla dahi entegre biçimde çalışabilmektedirler. Yapay zekanın uygulamadaki esnekliği sayesinde, algoritmalarının güncellendiği ve entegre edildiği donanımlarla sürekli uygulama geliştirme süreci devam etmektedir. Donanım ile entegre çalışabilen yapay zeka uygulamasına Maanyu vd.'nin Tesla marka aracın donanım çalışmaları örnek verilebilir [22]. Bu çalışma, yapay zekanın gelişimi ile beraber yapay zekanın gündelik hayata adaptasyonunun ne denli ilerlediğini dair bir örnektir. Çalışmada bahsedilen otopilot ile araçların tamamen kendiliğinden sürüş kabiliyetine sahip olması amaçlanmıştır. Otopilot geliştirilerek güvenli sürüş sağlayan akıllı araçlar oluşturulmuştur. Otopilot için geliştirmeler devam ederken, yapay zekayla entegre çalışan donanıma olan bağlılık giderek artmaktadır. Kendiliğinden sürüşü daha yüksek kabiliyetli geliştirme adına Nisan 2019'ta "Hardware 3" adı verilen donanım geliştirilmiştir. Hardware 3 tamamen Tesla'nın kendi üretimi olan bir işlemcidir. İlgili çalışma, yapay zekanın esnekliği ve yapay zeka geliştirmelerin uygun donanım kullanım sayesinde Tesla markasının, iyileştirmelere devam ettiği belirtilmektedir. Bu çalışmada da görüldüğü üzere yapay zekanın artık insanlarının gündelik hayatlarında yaygın kullandığı araçlardan biri olduğu söylenebilir. Böylece yapay zeka tasarımına bağlı olarak, gündelik hayatta özgün ve etkili çözümler üretilebilmektedir. Özgün problemlerde donanım ile entegre biçimde çalışabilecek yapay zeka çözümü için, MLP mimarisini ve çalışma sürecini incelemek gerekmektedir.

2.2.2 Yapay Sinir Ağlarının Temelleri

ANN, birden fazla algılayıcının (İng. perceptron) birbirleriyle ile olan bağlantılarından oluşmaktadır [6]. Her bir algılayıcı, girdiden ürettiği çıktı verisini bağlı olduğu diğer algılayıca sürerek, işlemin devamlılığını sağlar. ANN'de bulunan algılayıcılar paralel olarak çalışabilirler. Bahsedilen paralel çalışma yöntemi, biyolojik beyni taklit etme çabası olarak yorumlanabilir. Aşağıdaki görselde (Şekil 2.2), ANN'de bulunabilecek birden fazla girdiye sahip bir algılayıcının modeline ulaşılabilir.



Şekil 2.2. Temel sinir ağı modeli.

Algılayıcı diğer ismiyle düğüm, temel olarak, kendisine sürülen girdiyi sırayla birden fazla matematiksel denkleme tabi tutmaktadır. Bu matematiksel denklemler, temel anlamda yukarıdaki Şekil 2.2'den takip edilebilir. Öncellikle, her bir girdi (G) fonksiyonel olarak önceden kendisiyle eşleşmiş ağırlık (A) değeriyle çarpılır. Bütün çarpım sonuçları kümülatif toplanır. Devamında eğer var ise algılayıcının sapma değeri (B), kümülatif toplama eklenir. Elde edilen sonuç algılayıcı için önceden seçilen aktivasyon fonksiyonuna sürülür. Böylelikle algılayıcıya ait bir sonuç elde edilir. Temel sinir ağı modeli görüldüğü üzere birkaç matematiksel işlemden oluşmaktadır.

Aktivasyon fonksiyonu (f_N) algılayıcının ürettiği sonucu değerlendirip, karar veren temel algılayıcı elemanıdır [23]. Aktivasyon fonksiyonu, algılayıcının çıktı sonucunu ürettiği için aslında ANN yapılarının tahmin sonuçlarını büyük ölçüde etkiler. Tahmin sonucu

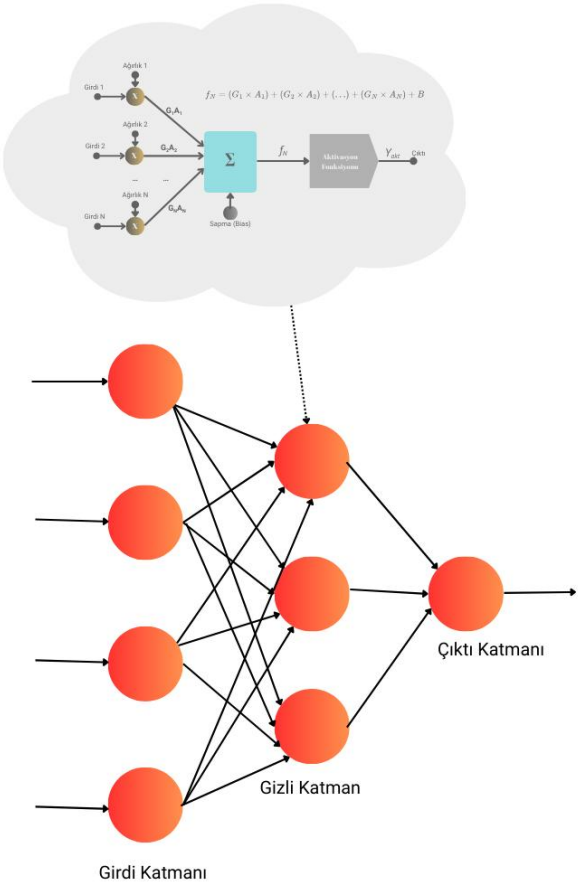
ürettiği için aktivasyon fonksiyonun seçimi de bir o kadar kritiktir. Aktivasyon fonksiyonun seçimi, ANN mimarisine ve kullanılan girdilere de bağlıdır [23].

Her bir algılayıcı aktivasyon fonksiyonu içermektedir. Aktivasyon fonksiyonu, kullanılan ANN modeline bağlı olarak doğrusal veya doğrusal olmayan fonksiyon olarak seçilebilir [23, 24]. Eğer aktivasyon fonksiyonu olarak basamak fonksiyonu kullanıldıysa, sıfır gradyan ile karşılaştıklarında, basamak fonksiyonu, gradyan değerlerinde güncelleme yapamadığı için algılayıcı sonuca yakınsayamayacaktır. Bundan dolayı bu problem için basamak fonksiyonu yerine doğrusal fonksiyon kullanılabilir [24]. Fakat doğrusal fonksiyon tercihinde ise (-∞, ∞) arasında gradyan güncellenmesi geri dönülemeyecek değerlere çıkabilir. Bu tarz problemler ile karşılaşmamak için sigmoid veya softmax gibi doğrusal olmayan fonksiyonlar kullanılabilir [24].

Çizelge 2.2. ANN için aktivasyon fonksiyonu kullanım önerileri [23,24].

Fonksiyon	**Yorum**	**Ne zaman Kullanılır?**
Basamak	Geri-yayılımlı algoritma ile çalışılmıyorsa tercih edilir.	Genelde tercih edilmez.
Sigmoid	İkili sınıflandırma problemlerinde tercih edilir.	Boole (İng. boolean) cebrine benzetimi yapılamamaktadır.
ReLU	Gizli Katmanlar için en popüler yöntemdir.	Sıklıkla tercih edilir.
Softmax	Birden çok sınıf bulunan modellerde, tek sınıfın tahmini için tercih edilir.	Çoklu Sınıflandırmadan tek sınıfı tahmin etme durumunda tercih edilir.

Aktivasyon fonksiyonun seçimi, algılayıcının tahmin sonucunu büyük ölçüde etkilemektedir. Çizelge 2.2 incelenerek bazı aktivasyon fonksiyonların ne zaman kullanılacağına dair çıkarımlara ulaşılabilir. Fakat karmaşık problemleri çözümlemek için tek algılayıcı ve ona ait aktivasyon fonksiyonu yetersiz kalmaktadır. Tek algılayıcının yetmediği durumlarda birden fazla algılayıcı içeren ve birden çok katmana sahip algılayıcı modeli yani MLP modeli kullanılır [25].



Şekil 2.3. Örnek çok katmanlı algılayıcı modeli.

İleri besleme MLP modeli Şekil 2.3'te temsili olarak verilmiştir. Görüldüğü üzere her bir algılayıcı aslında Şekil 2.2'deki mimariyi içerir. Fakat bir algılayıcı, birden fazla algılayıcı ile bağlantı oluşturarak kendisine sürülen bilgiyi işler ve ürettiği bilgiyi de sıradaki algılayıcıya sürer. Böylece, karmaşık problemleri çözmek için birden fazla katman içeren MLP modeli tercih edilir [25].

MLP modelleri, yapay zeka uygulamalarında sıklıkla tercih edilen modellerden biridir. Çünkü tek algılayıcının, karmaşık problemleri çözümlemek için yetersiz kalmaktadır. MLP modeli ise içerdiği birden çok ve süregelen algılayıcı bağlantıları sayesinde tahminin doğru sonuca yakınsama olasılığını artırır. MLP modeli, ticari kullanımlardaki birçok DL veya ML modelinin de temelini oluştururlar [25]. Örneğin, evrişimsel sinir ağları (İng. Convolutional Neural Network-CNN) genelde fotoğraflardan veya resimlerden görsel tanımlama için kullanılır ve bu sinir ağları aslında MLP'nin bir tür özelleştirilmiş modelidir

$$f(x) = f_{çıktı}\left[f_{gizli}(f_{girdi}(x)\right] \quad (2.1)$$

Denklem 2.1'de, Şekil 2.3 incelenerek oluşturulan bir zincir fonksiyonu görülmektedir. Fonksiyon isimlerinden, katmana ait fonksiyon takip edilebilir. MLP modelinin bütün boyutunu, zincir fonksiyonunun derinliği belirler. "Derin Öğrenme" teriminin ortaya çıkışı bu yaklaşımla ilişkilendirilebilir [26]. MLP mimarileri genelde üç temel katmandan oluşur. Girdilerin sürüldüğü katmandan itibaren katmanların isimleri sırasıyla: Girdi katmanı, gizli katman ve çıktı katmanıdır. Girdi katman, isminden de anlaşılacağı üzere, dış dünya ile kurulan bağlantı üzerinden kendisine sürülen girdileri işler. MLP modellerinde girdi katman sayısı genelde bir tercih edilir. Gizli katman, girdi katmandan gelen veriyi işleyerek, çıktı katmanına kadar sürülmesini sağlar. Toplam katman sayısı, problemin karmaşıklığına bağlı olarak artırılabilir. Çalışmalar sonucunda halen kaç tane katman ve düğüm kullanılması gerektiğine kesin cevap verilememektedir [27]. Fakat probleme bağlı olarak kullanılması gereken girdi katmanındaki düğüm sayısı, çıktı katmanındaki düğüm sayısına bağlı olarak gizli katman miktarı ve gizli katmanda kullanılması gereken düğüm sayısı için aralık belirlenebilmektedir [27].

MLP modeli, daha da özelleştirilmiş modellerine kıyasla daha az hesaplama yükü ile karmaşık problemleri sonuca yakınsama ve DL ile ML'de kullanılan çoğu modelin de temel yapıtaşını oluşturduğu için donanım tanımlama dili (İng. Hardware Description Language-HDL) kullanılarak yapılan gerçekleştirmelerde tercih edilir. Tez çalışmasında, literatür araştırması sonucunda hareket tanıma için MLP kullanarak yüksek doğruluk değerlerine ulaşılabileceği sonucu çıkarılmıştır [28]. İlgili çalışmada yinelemeli sinir ağların finansal tahminler, çeviriler ve ses tanımlarında başarılı olduğundan bahsedilmiştir. Çalışmada CNN yerine kendi tabirleriyle derin sinir ağları (İng. DNN) tercih etmelerinin nedeni ise FPGA'te kaynak tüketiminde kayda değer farklılıkta gerçek zamanlı görüntü sınıflandırma çalışmasında yüksek başarı kaydettiklerini ifade etmişlerdir. İlgili çalışmada aktivasyon fonksiyonu olarak da ReLU ve Softmax kullanarak görüntü tanımlayıp, sınıflandırmada %99 gibi yüksek bir başarı yüzdesine ulaştıklarını test sonuçları ile beraber sunmuşlardır. Ek olarak bu başarıyı, çalışmalarında kıyasladıkları aynı amacı benimseyen diğer çalışmalara kıyasla çok daha düşük kaynak tüketimi ve daha yüksek doğruluk (karşılaştırmak için kullandıkları diğer çalışmaların doğruluk yüzdeleri sırasıyla: %91 ve %89'dur) yüzdesine ulaştıklarından bahsetmişlerdir. İlgili çalışmada girdi katmanında ve ilk gizli katmanda 30'ar düğüm ve ikinci gizli katman ile çıktı katmanında 10'ar düğüm olmak üzere toplam 80 düğüm kullanmışlardır. Tez

çalışmasında da kullanılan toplam düğüm sayısı aynıdır. Tez çalışmasında MLP kullanımının tercih edilmesinin nedenleri, literatürde bu çeşit uygulamalar için daha verimli kaynak kullanımı ve yüksek doğruluk değerlerinin ulaşılmış olmasıdır.

2.3. Yapay Zeka Uygulamaları için Yonga Tasarımı

2.3.1 Literatürde Yapay Zeka Uygulamaları için Kullanılan Donanımlar

Yapay zeka yazılımlarının gerçeklenmesi için veri setlerinin büyüklüğü ve algoritmaların karmaşıklığından dolayı, her geçen gün daha yüksek kapasiteli donanımlara ihtiyaç duyulmaktadır [29]. Tümleşik devreler, yapay zeka uygulamalarının yüksek işlem maliyetlerini, düşük güç tüketerek gerçekleştirebilirler. Bu tümleşik devreler, literatürde yapay zeka yöntemlerini hızlandıran donanımlar olarak da bilinirler [30]. D.D. Prashant vd. [31] genel amaçlı sinir ağı modeli tasarımında donanım tercihinin ne denli kritik olabileceğini cevaplandırmaya çalışmışlardır. İlgili çalışmada MLP'nin, biyolojik sinir ağını taklit etmek için paralel çalışma yapısından ötürü yoğun hesaplama isterine ihtiyaç duyduğuna değinilmiştir. Böylelikle MLP mimarisinin donanım üzerinde gerçekleştirerek şekil tespit, tanıma ve sınıflandırma problemlerini başarılı biçimde sonuçlandırabileceği belirtilmiştir. İlgili çalışmada sunulan Çizelge 2.3, tez çalışmasının kapsamında değerlendirmeye alınmıştır.

Çizelge 2.3. ANN tasarımının parametrelere bağlı farklı donanımlarda karşılaştırmaları [31]

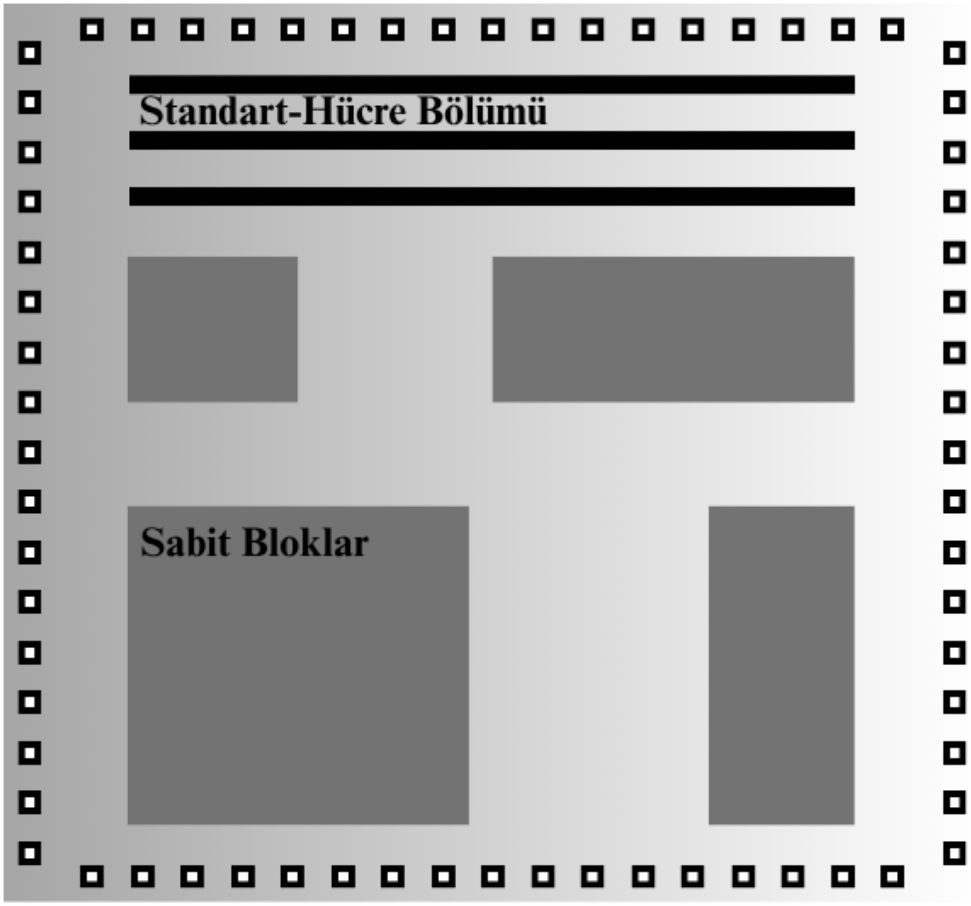
Parametre	**Analog ASIC**	**Sayısal ASIC**	**FPGA**	**İşlemci**
Kullanılan Alan	+ + +	+ +	+	-
Hız	+ + +	+ +	+	-
Maliyet	- -	- -	+ +	+ +
Tasarım Süresi	- -	- -	+	+ + +
Güvenilirlik	- -	+	+ +	+ +

Çizelge 2.3'te Analog ASIC'nin kullanılan alan ve hız konusunda oldukça avantajlı olduğunu fakat maliyet, tasarım süresindeki dezavantajları ve gürültüden etkilenmesinden kaynaklı güvenilirlik konusunda da oldukça dezavantajlı olduğu görülmektedir. FPGA yerine VLSI kullanmanın daha uzun tasarım süreci ve çok daha

yüksek maliyetler gerektirdiğinden bahsedilmiştir. Tez çalışmasında, sayısal tasarım hem FPGA'da hem de ASIC olarak gerçeklendirilmiştir. MLP mimarileri yüksek seviyede hesaplama ihtiyacından dolayı yüksek güç tüketimi ve yüksek sayıda işlem kapasitesine ihtiyaç duymaktadır. Bunun için MLP modelinin sayısal tasarımında çok sayıda mantıksal elemana ihtiyaç duyulmaktadır [31]. Sayısal tasarım, FPGA ile gerçekleştirilirse daha kısa süre tasarım süresi ve tekrardan konfigüre edilebilme avantajına sahiptir. Fakat FPGA yerine ASIC kullanılarak MLP tasarımı daha düşük güç tüketimi ve birim alan başına daha az mantıksal eleman ile daha yüksek hızlı tasarım sonuçları sunabilir [32]. Karmaşık problemlere daha etkin çözümler getirmek için FPGA ile doğrulanmış tasarımların ASIC'ye aktarılması yöntemi özellikle ticarileşecek yongalar için maliyet etkin bir yaklaşım da sunabilmektedir. Literatür araştırmaları sonucunda, tez çalışmasında öncellikle MLP mimarisi yüksek seviye dil yazılımı ile bilgisayarda doğrulanmıştır. MLP mimarisi doğrulandıktan sonra sayısal tasarıma aktarılarak, FPGA üzerinden benzetim ile donanım seviyesinde doğrulanmıştır. Son olarak, tasarım ASIC üretimine uygun RTL2GDSII biçimine dönüştürülmüştür. Böylelikle literatür araştırmalarının da sunduğu düşük güç tüketimi, daha yüksek hızı ve birim başına düşen eleman sayısı gibi parametreler değerlendirilerek, MLP mimarisi donanım üzerinde gerçekleştirebilecektir. Yapay zekayı ivmelendirici donanımlar üzerinde halen araştırmacılar araştırmaları ile literatüre katkı sunmaya devam etmektedirler [32, 33].

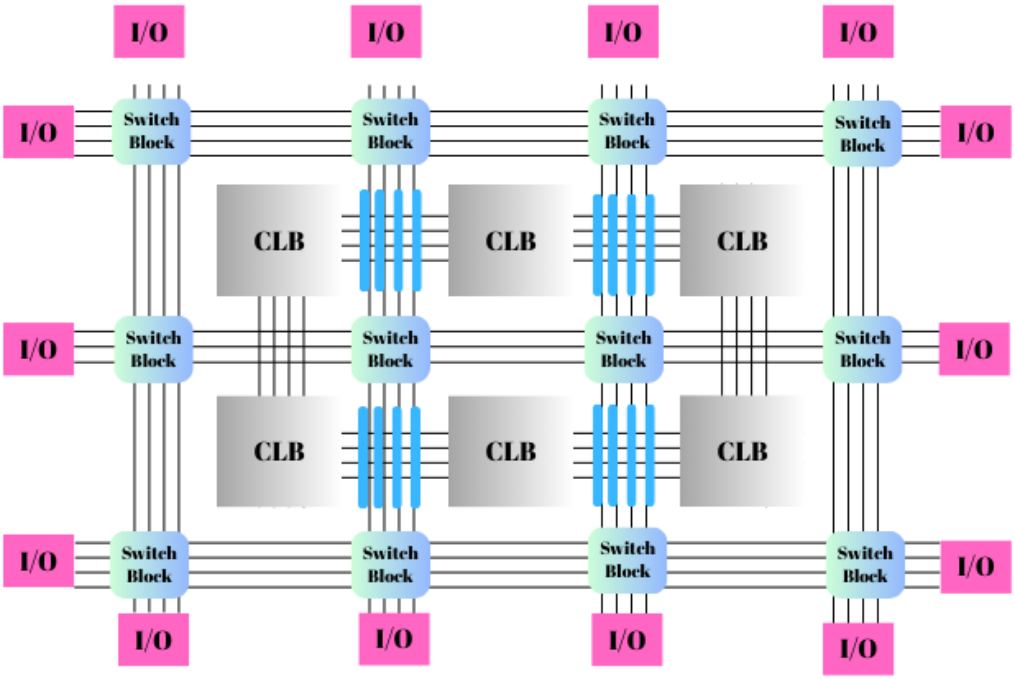
2.3.2 Kullanılacak Donanımlar Hakkında Genel Bilgiler

Tümleşik devreler, içerdikleri alt devreler sayesinde üstlendikleri görevleri denetimli biçimde sürdürebilirler. Tümleşik devreler, özgün amaçlı görevleri yerine getirenlere ASIC ve ASIC olmayan olarak iki genel sınıfta incelenebilirler [34]**.** Tasarım isterlerine bağlı olarak hangi aygıtın seçileceği parametrelerin titiz biçimde belirlenip değerlendirilmesiyle karar verilmesi gerekir.



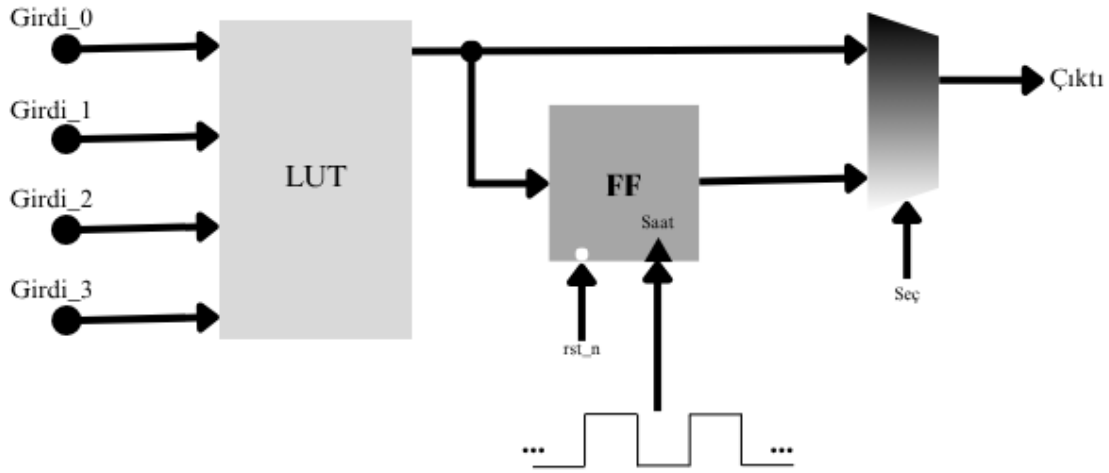
Şekil 2.4. Standart hücrelerden oluşan ASIC mimarisi.

Şekil 2.4'te standart hücrelerden oluşan ASIC yapısına ait bloklar görülmektedir. Şekil 2.4'teki görsel tasarlanırken V. Tarate'nin kitabından faydalanmıştır [35]. Bu bloklarda önceden hazırlanmış kütüphaneler kullanılabilir. Sabitlenmiş blokların kullanımı, tamamen özel tasarım ASIC'lere kıyasla tasarım süresinden kazanç sağlar fakat tamamen özel tasarımlarda, kullanım amaçlarına özgün tasarım yapmanın da ayrı avantajları (düşük güç tüketimi ve daha yüksek hızlı mimari) bulunmaktadır [35]. Günümüzün dinamik teknolojik gelişmeleri ve yeniden şekillenen ihtiyaçlardan ötürü FPGA'ler akademik ve endüstriyel çalışmalarda kendilerine yer edinmektedirler.



Şekil 2.5. Kavramlar üzerinden örnek FPGA mimarisi.

Şekil 2.5'de FPGA'da bulunan elemanların konumlanması ve bağlantılarını içeren örnek bir mimari görülmektedir. Görsel hazırlanırken V. Taraate'nin kitabından [35] faydalanılmıştır. FPGA, büyük çoğunluğu ara bağlantılar ve anahtarlama bloklarından oluşan, tekrardan konfigüre edilebilir yapısı ve hızlı prototipleme kabiliyeti sayesinde tasarımların daha kısa sürede tamamlanmasında önemli rol oynar [36]. Şekil 2.5'de görüleceği üzere FPGA, mantıksal işlemleri gerçekleştirmek için içerisinde CLB (İng. Configurable Logic Block) yapılarını bulundurur [37]. CLB yapısı da her ne kadar FPGA mimarisi gibi, FPGA teknolojisine, üretim yöntemine ve üretimini gerçekleştiren firmaların bağlı oldu mimariye bağlı olsa da en temel haliyle başvuru çizelgesi (LUT) ve çıktı yazmacı içerir [36]. Genellikle CLB yapısına MUX da eklenir. Böylelikle anlık LUT çıktısı mı yoksa FF çıktısı mı kullanılacağı denetlenebilmektedir [37]. Denetim ağının artırılması, tasarımcının kombinasyonel (İng. combinational) veya ardışıl (İng. sequential) tasarımları arasındaki bağlantıyı artırır ve iki tasarım arasındaki geçişi kolaylaştırır. Şekil 2.6'da 4-girdiye sahip LUT bloğu bulunan örnek CLB mimarisi görülmektedir [36].



Şekil 2.6. Kavramlar üzerinden örnek 4-girdi CLB mimarisi.

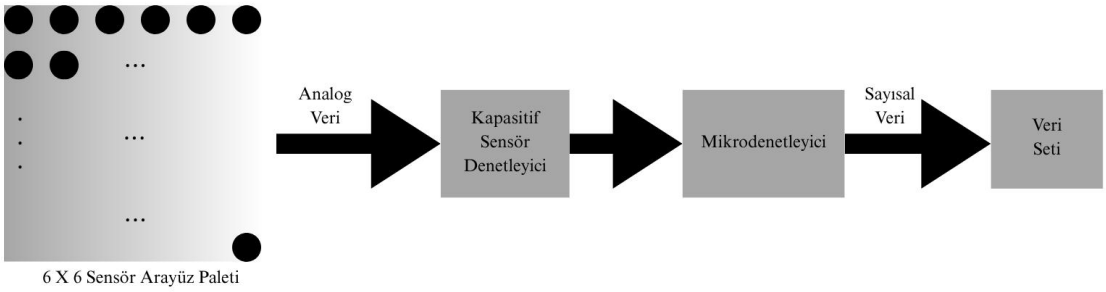
LUT, sentez sonucu sentezleyicinin, tasarıma sürülen girdilere ve istenilen çıktılara bağlı olarak elde edilen boole (İng. boolean) cebirini saklayan yapılardır. Böylelikle tasarımın bütün kombinasyonları tek bir aygıt üzerinden yönetilebilir. LUT çıktısını muhafaza ederek, istenildiğinde tasarım içerisinde kullanılmasını sağlar. LUT tabanlı hücreleri içeren FPGA'ler, bu teknoloji sayesinde farklı davranışlara sahip mantıksal fonksiyonları programlayabilir veya konfigüre edebilirler [35]. Fakat LUT mimarisi FPGA teknolojisine ve kullanılan teknolojiye ait sentezleyiciye bağlı olabilir [36]. Teknoloji bağımlılığından kaynaklı belirsizlikler kaçınılmaz olup aynı sayısal tasarım, farklı FPGA mimarilerinde farklı kaynak tüketim sonuçları doğuracaktır. Bu belirsizlikler ve farklı kaynak tüketimlerini karşılaştıran literatür araştırmaları ve kitaplar bulunmaktadır [34, 36].

3. TASARIMLAR VE DENEYSEL ÇALIŞMALAR

3.1. Tez Çalışmasında Kullanılacak Sistemler

Tez çalışmasında model tanıma ve tespiti gerçekleştirmek için birtakım sistemler tasarlanmıştır. Literatür araştırması sonucunda, model olarak el hareketleri uygun görülmüştür [13, 19, 20]. El hareketi tanıma ve tespiti için veri seti oluşturulması veya hazır veri seti kullanılması gerekir. Veri seti oluşturmak için CPS dizilerinden oluşan bir palet tasarlanmıştır. Sensör paletinde, sensörlerin arasındaki mesafe eş ve her bir satırda aynı sayıda bulundurularak el hareketinin her bir sensörde eş seviyede dağıtılmıştır [6]. Tez çalışması süresince, 16, 25 ve 36 tane CPS bulunduran sensör dizileri kullanılmıştır. Sensör arayüz paletinin içerdiği CPS miktarının kararlaştırılması 3.2.1. Veri İşleme için Yapay Zeka Tasarımı başlığı altında sunulmuştur.

Veri seti oluşturmak için öncellikle, el hareketinin sensör arayüz paletine yaklaştırılması gerekmektedir. Palete yaklaştırılan el hareketi, CPS dizilerinin analog veri oluşturmasını sağlar. Analog veri, kapasitif sensör denetleyici ile mikrodenetleyiciye aktarılır. Sayısal veri, gerek duyulduğunda mikrodenetleyicide işlemlere tabi tutulduktan sonra veri seti oluşturmak için kaydedilir. Veri seti oluşturma akışı Şekil 3.1'de sunulmuştur.

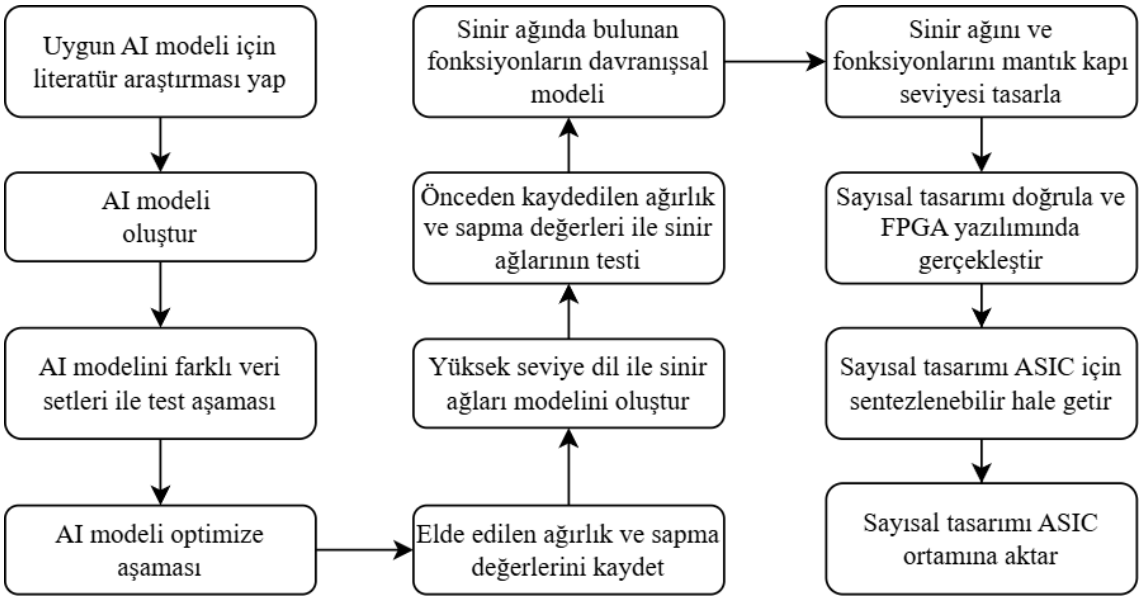


Şekil 3.1. Veri seti oluşturma akışı.

Şekil 3.1'den de takip edileceği üzere yaklaştırılan el hareketleri ile ham veri seti oluşturulmuştur. Ham veri seti, metin belgesi (İng. text file types-txt) formatında kaydedilmiştir. Ham veri seti, yapay zekanın eğitimi için kullanılacağından dolayı ön işlemden geçirilmesi gerekir. Bu çalışmalara 4. Bölüm'de yer verilmiştir. Veri seti oluşturma aşamasından sonra veri işleme aşaması başlamaktadır. Veri oluşturma aşaması sensör arayüz paleti sistemini kapsarken, veri işleme geri kalan sistem bütününü kapsar.

Tez çalışmasında veri seti oluşturulmadan önce AI modelinin olgunlaşması sağlanmıştır. AI modelinin çıktı analizleri için hazır veri setleri kullanılmıştır. Çünkü veri işleme sistemi için gerekli parametreler (veri seti boyutu ve veri tipleri), AI modelinin çıktılarına göre belirlenmiştir.

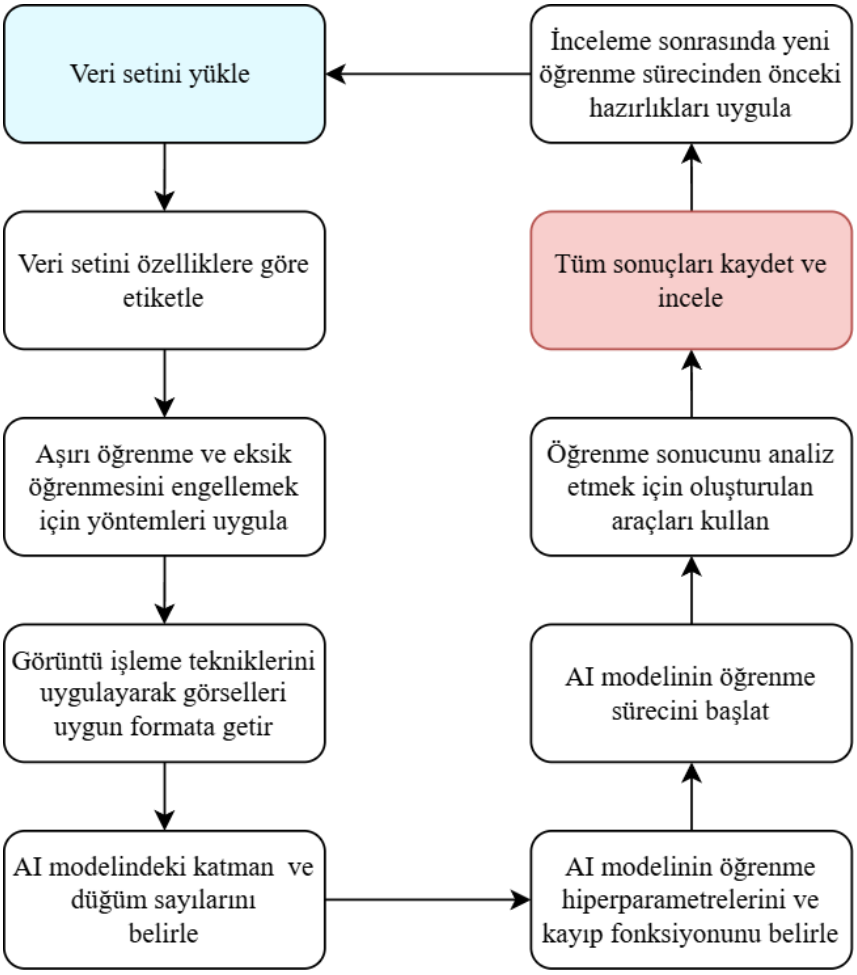
3.2. Veri İşleme Sistemi



Şekil 3.2. Tez çalışması için veri işleme sistemi süreç akışı.

Şekil 3.2'de tez çalışması için tasarlanan, veri işleme sisteminin süreç akışı görülmektedir. Akış, en sol yukarıdaki bloktan başlar ve en sağ aşağıdaki blok ile sonlanır. Başlangıçta, tez çalışmasında kullanılacak AI modeli için literatür araştırması yapılmıştır [11-13, 18-21]. Literatür araştırması sonucunda benimsenen AI modelinin iteratif öğrenme süreci Şekil 3.3'te sunulmuştur. İteratif öğrenme süreci, modele yüklenen veri seti ile başlamıştır (mavi kutu). AI modelinin öğrenme süreci sonucu olgunlaşması için farklı hazır veri setleri kullanılmıştır [38-42]. Hazır veri setleri ile ilgili çalışmalara 3.2.1.1. Veri işleme sistemi için derin öğrenme uygulaması isimli başlıktan ulaşılabilir. Hazır veri setleri seçilirken benzer tipte veri içermelerine dikkat edilmiştir [38-42]. Hazır veri setleri, özelliklerine göre etiketlendirildikten sonra iteratif öğrenme süreci başlatılmıştır. İteratif öğrenme süreçlerinin sonucu analiz edilerek, en uygun hale getirilmiştir. AI modelinin en uygun haline, kullanılan yöntemler ve hiper parametrelerin öğrenme aşamasında güncellenmesiyle ulaşılmıştır. En uygun model, denektaşı testi (İng. benchmarking test) sonucunda belirlenmiştir. Öğrenme sürecinde AI modelin ağırlık ve

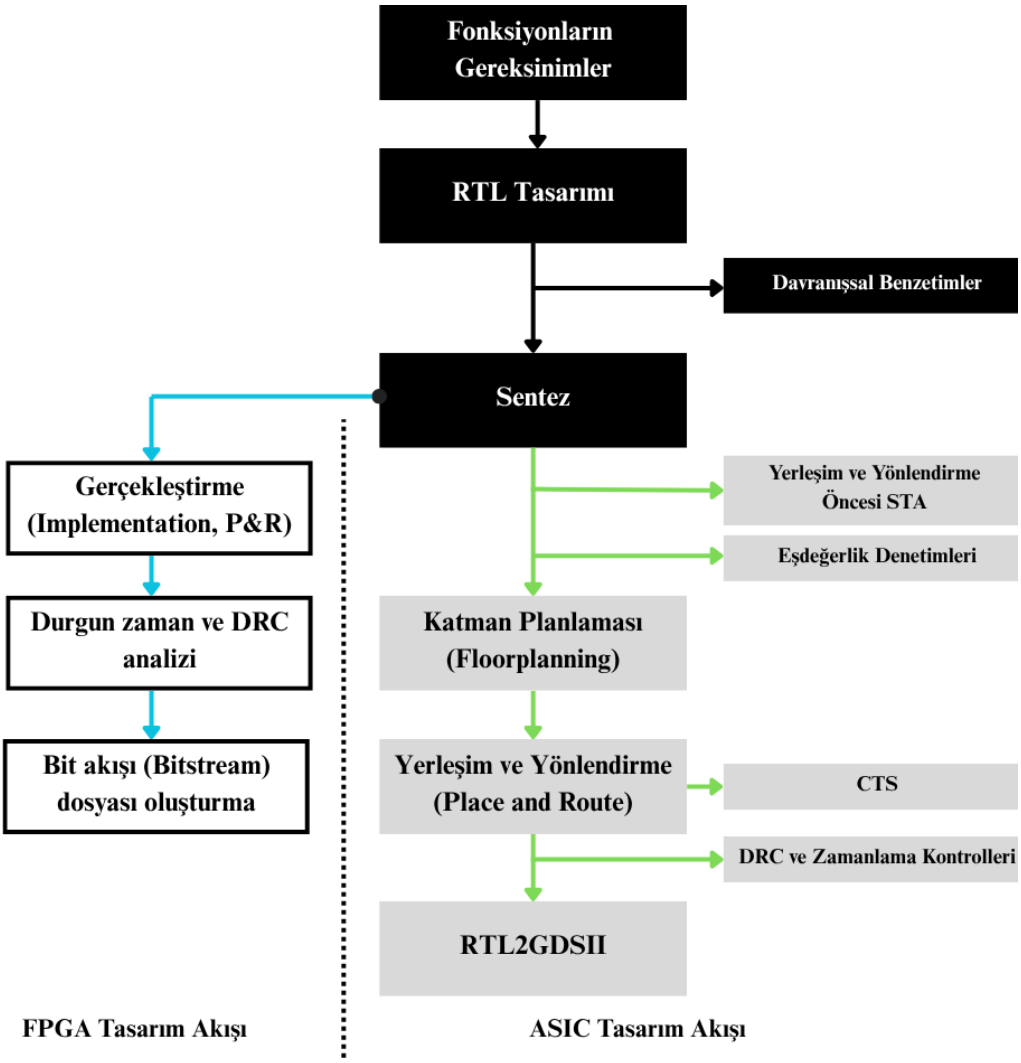
sapma değerleri kaydedilmiştir. En uygun modelin, ağırlık ve sapma değeri uygun kullanım formatına dönüştürülmüştür. Uygun formattaki ağırlık ve sapma değerleri yüksek seviye dil ile tasarlanacak MLP modelinde ve sayısal tasarımda kullanılmıştır. Yüksek seviye dil ile tasarlanmış MLP modeline, önceden eğitilmiş modelin ağırlık ve sapma değerleri sürülerek MLP mimarisi test edilmiştir. MLP mimarisi test edildikten sonra sayısal tasarıma aktarılmaya başlanmıştır. Böylelikle RTL seviyesindeki MLP mimarisinin, sayısal tasarım süreci hızlandırılmıştır [32].



Şekil 3.3. AI modelinin iteratif öğrenme süreci.

Veri işleme sisteminin süreç akışında aslında ASIC tasarımına kadar olan her bir adımda MLP modeli test edilip, doğrulanmıştır. Bu sayede MLP mimarisinin, sayısal tasarımına başlamadan önce karşılaşabilecek problemler belirlenmiştir. Bu süreç tercihi, her ne kadar

tasarım sürecini uzatsa da tasarım sürecinde planlı ilerlemeyi ve beklenmedik problemleri öngörmeyi sağlamıştır.



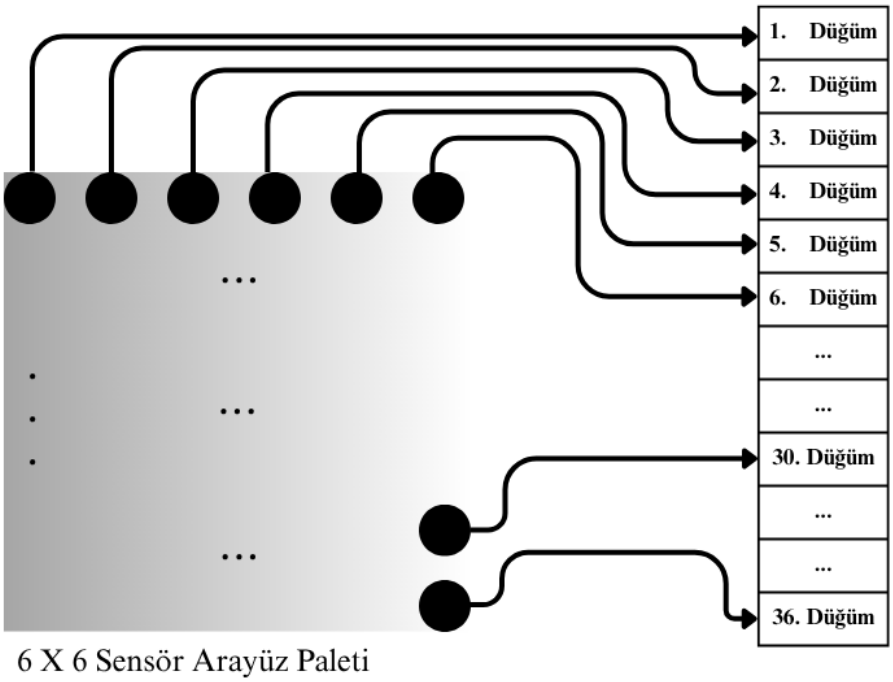
Şekil 3.4. FPGA ve ASIC için tasarım akışı.

Şekil 3.4'te FPGA ve ASIC için literatürden [34, 35] faydalanılarak oluşturulan tasarım akışı görülmektedir. Şekil 3.4'teki siyah renkli bloklar FPGA ve ASIC için ortak tasarım sürecini göstermektedir. Beyaz renkli bloklar sadece FPGA'ya ait tasarımı akışını ve gri renkli bloklar ise sadece ASIC'ye ait tasarımı akışını belirtmek için kullanılmıştır. MLP mimarisinin sayısal tasarıma aktarılmasında, HDL için Verilog tercih edilmiştir. MLP mimarisinin sayısal tasarımı için 3.2.1.3. Veri işleme sistemi için sayısal tasarım başlığı incelenebilir. Şekil 3.4'te FPGA ve ASIC için olan tasarım akışının ayrıldığı süreçler görülmektedir. ASIC tasarım akışına ait çalışmalar 5. Yapay Zeka Yongası başlığı altında

bahsedilmiştir. MLP mimarisi hem doğrusal hem de doğrusal olmayan aktivasyon fonksiyonları içerebilir [43]. MLP mimarisinin sayısal tasarımında en uzun tasarım süreci doğrusal olmayan aktivasyon fonksiyonların incelenmesi ve tasarlanması olmuştur. Bu incelemeler, 3.2.1.3. Veri işleme sistemi için sayısal tasarım başlığı altında bahsedilmiştir. Görüldüğü üzere, MLP mimarisinin sayısal tasarımı için bütün yapıtaşlarının değerlendirilmesi gerekir. Çünkü sayısal tasarım en sonunda, ASIC seviyesine aktarılacağı için her yapının sayısal donanım seviyesinde tasarlanabilmesi gerekir. Şekil 3.4'teki tasarım akışının son aşaması ile Şekil 3.2'de bulunan veri işleme sistemi süreç akışının son kısımları ortaktır. Böylelikle, son aşama ile artık veri işleme sistemi süreç akışı sonlanmış olur.

3.2.1 Veri İşleme Sistemi için Yapay Zeka Tasarımı

Şekil 3.3'te tasarlanacak AI modeline en uygun yapının bulunması için takip edilen iteratif öğrenme süreç akışı görülmektedir. Süreç akışında bahsedilen veri setini oluşturmak için CPS dizileri kullanılmıştır. 2. Genel Bilgiler başlığında tartışıldığı gibi, yapay zeka tasarımında MLP mimarisi kullanılmıştır. MLP'nin girdi katmanında bulunan her bir düğümün dizinin her bir sensörüne bağlanması gerekir. Çünkü girdi katmanındaki her bir düğüm tek girdiye sahiptir.



Şekil 3.5. CPS'den elde edilen verilerin girdi katmanı düğümlerine sürülmesi.

Şekil 3.5'te her bir CPS'nin, girdi katmanında bulunan her bir düğüm ile arasındaki bağlantıyı göstermektedir. Nihai tasarımda 36 CPS ve dolayısıyla girdi katmanında da 36 düğüm kullanılmıştır. Deneysel çalışmalar sırasında ve AI modelini en uygun hale getirmek için açık kaynak veri setleri araştırılmıştır. Açık kaynak veri setlerinin tez çalışmasına uygun olmasına dikkat edilmiştir [38-42]. Bu veri setleri görüntü verilerini içerdiği için AI modelinin oluşturulması sürecinde sadece öncül çalışmalarda kullanılmıştır. Seçilen veri setleri için Şekil 3.3'te yer verilen iteratif öğrenme süreci uygulanmıştır. İteratif öğrenme süreci, AI modeli en uygun sonuçları üretene kadar devam etmiştir. Algoritmada ilk olarak Şekil 3.3'te bulunan açık mavi renkli kutucukta yazıldığı gibi kullanılacak veri seti seçilir ve yüklenir. AI modeli kendisine yüklenen veri setleri üzerinden sınıflandırma yapması gerektiğinden veriler öz niteliklerine göre etiketlenmiştir. Sınıflarda az sayıda veya eşit dağılmamış veri seti bulunursa AI modeli öğrenme sırasında, çok fazla parametre kullanacağından dolayı öğrenmenin başarısız olma ihtimali artmaktadır [26, 43]. Şekil 3.5'te gösterildiği gibi veri setlerinde bulunan yapıda yer alan her bir CPS'nin değeri, girdi katmanındaki düğümlere sürülecektir. Tezde yer alan CPS değerleri geleneksel görüntü ile özdeşleştirilmesi amacıyla piksel olarak da ifade edilmiştir. Bundan dolayı belirlenen MLP mimarisine uygun boyutlarda veri seti ile çalışılması gerekir. Fakat hazır veri setleri incelendiğinde her birinin ayrı piksel boyutunda toplandığı anlaşılmaktadır [38-42]. Veri setlerini uygun boyutlara getirmek için kullanılabilecek yöntemlerden biri PCA (İng. Principal Component Analysis) tekniğidir [26]. Diğer yöntem ise alt-örnekleme uygulamasıdır [44]. Alt-örnekleme ve PCA yöntemi kullanılarak ön işleme uygulaması veri setindeki görsellerin sonucunda boyutları düşürülmüştür. Böylece AI modeli, girdi katmanında 36 düğüm ile çalışabilmiştir. Görseller üzerinden anlam çıkarmak için AI modelinin görüntü işlemesi gerekmektedir. AI modelinin karmaşık çözümlemeleri azaltarak işlem maliyetini düşürmek için görüntüleri gri formatına dönüştürmek ve düzleştirmek literatürde kullanılan yöntemlerdendir [45, 46]. CPS dizisi ile toplanan veri tek boyutlu olduğu için, tez çalışmasında gri ölçekli görseller ile çalışılmıştır. Siyah beyaz yerine gri ölçekli görseller ile çalışarak AI modelinin daha fazla bilgi ile öğrenmesi sağlanmıştır. Ek olarak 6x6 piksel Şekil 3.5'te gösterildiği gibi 36x1 formatına çevrilmiştir. Bu işleme literatürde düzleştirme (İng. flatten) denir. Böylece AI modeli, tek girdi katmanından bütün girdi verilerini içerebilir [46]. Bu yöntemler uygulanarak MLP mimarisinin, 2. Bölümde bahsedilen literatür araştırmalarındaki gibi uygun maliyette donanım kullanılarak yüksek doğruluk değerine ulaşılabileceği öngörülmektedir. Şekil 3.3'te bulunan akış takip

edildiğinde sonraki aşamalarda katman ve düğüm sayısını belirlemek olacaktır. Literatürde, MLP modelinin katman sayısı ve düğüm sayısının nasıl belirlenebileceğine dair önerilen bazı hesaplama yöntemleri bulunmaktadır [27, 47]. Tez çalışmasında AI modeli, sayısal tasarıma aktarılacağı için olabildiğince az sayıda düğüm sayısı ile yüksek doğruluk değerlerine ulaşılması istenmektedir. Literatürde gizli katmandaki düğüm sayısının, girdi ve çıktı katmanları ile ilişkisi:

$$N_{Gizli\ Katman} > \frac{2}{3}\ N_{Girdi\ Katmanı} + N_{Çıktı\ Katmanı} \quad (3.1)$$

$$2 \times N_{Girdi\ Katmanı} > N_{Gizli\ Katman} \quad (3.2)$$

MLP modelinin düğüm sayısı Denklemler 3.1 ve 3.2'de bulunan formüller kullanılarak belirlenmiştir [47]. *N*, ait olduğu katmandaki düğüm sayısını belirtir. Belirlenen MLP modelinin düğüm sayıları sırasıyla 36, 20, 20, 4'tür. Denklem 3.1'in sol tarafı 40 iken, denklemin sağ tarafı 28'dir; Denklem 3.2'nin sol tarafı 72 iken, denklemin sağ tarafı 40'tır. Böylelikle formüller üzerinden tez çalışması için belirlenen düğüm sayısının uygunluğu da doğrulanmıştır [47]. Şekil 3.3'teki akış iteratif biçimde çalıştırılmıştır. AI modeli koşturulurken belirlenen hiper parametreler elde edilen sonuçlara göre güncellenmektedir.

Çizelge 3.1. AI modeli için nihai hiper parametre ve değişken değerleri.

Değişken	**Değeri veya Türü**
MLP Mimarisinin Katman Sayısı	4
MLP Mimarisinin Soldan Sağa Düğüm Sayısı	36 / 20 / 20 / 4
Katmanlarda Kullanılan Aktivasyon Fonksiyonu	- / ReLU / ReLU / Softmax
MLP Mimarisinin Toplam Parametre Sayısı	1244
Toplu Boyut (İng. batch size)	30
İterasyon (İng. epoch)	35
Öğrenme Oranı (İng. learning rate)	0,00005
Optimize Edici (İng. optimizer)	Adam
Kayıp Fonksiyonu (İng. loss function)	Kategorizelenmiş Çapraz Entropi
Veri Setinin Dağılımı (Eğitim, Doğrulama, Test)	%75 / %15 / %10

3.2.1.1 Veri İşleme Sistemi için AI Modeli Çalışmaları

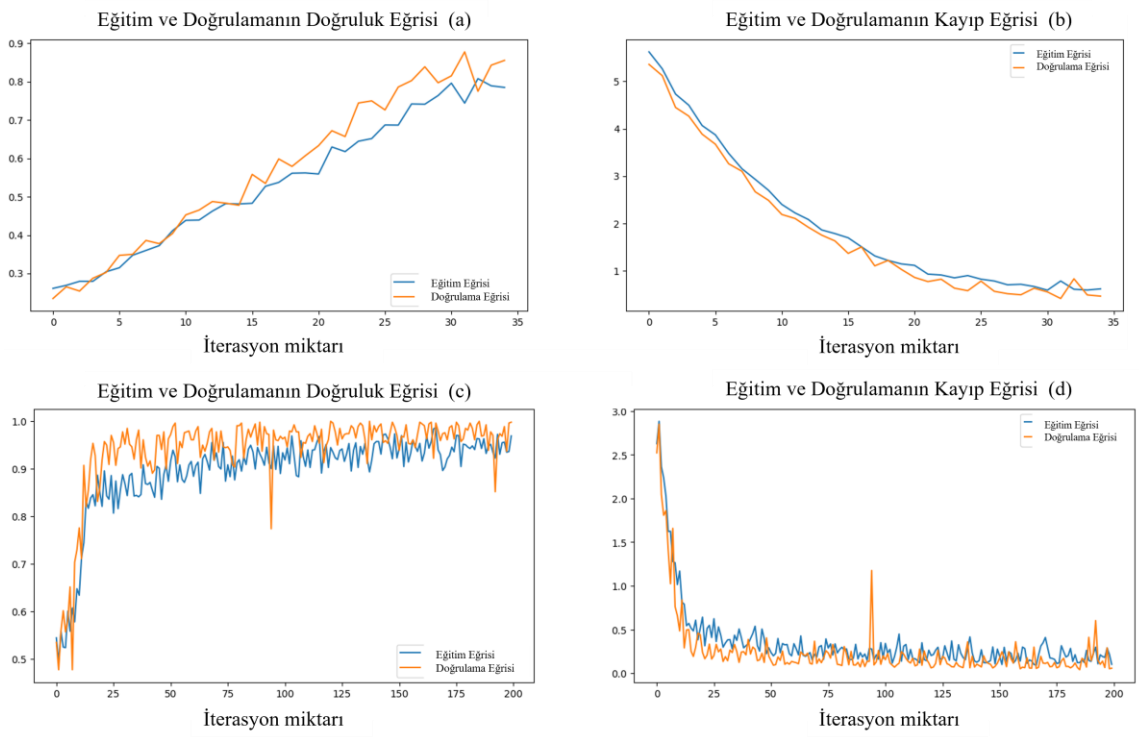
AI modeli çalışmalarının öncül aşamasında hazır veri setleri kullanılmıştır [38-42]. Bu veri setleri doğrudan CPS verisi olmamakla beraber sadece öncül çalışmalarda AI modelini olgunlaştırmak için kullanılmıştır. Şekil 3.3'te bulunan akış takip edilerek iteratif denemeler sonucunda AI modelinde belirlenen hiper parametreler ve değişkenler Çizelge 3.1'te verilmiştir. Değişkenlerin kararlaştırılması için Şekil 3.3'teki akış takip edilmiştir. İlk denemelerde MLP mimarisinde girdi katmanındaki düğüm sayısı 16, 25 ve 28 olarak belirlenmiştir. Doğruluk değerleri ve F1 puanları düşük geldiği için girdi katmanındaki düğüm sayısı 36'ya çıkarılmıştır. Çünkü modelin doğruluğundan emin olmak istenmiştir. Hem PCA kullanılarak hem de girdi katmanında 28 düğüm kullanılarak elde edilen sonuçlar aşağıdaki çizelgede (Çizelge 3.2) verilmiştir.

Çizelge 3.2. Seçilen veri seti [38] kullanılarak PCA ile elde edilen sonuçlar.

İşlem Sırası	**Öğrenme Oranı**	**Toplu Boyut**	**İterasyon**	**Veri Setinin Eğitim Doğruluk Oranı**	**Kayıp Fonksiyonu**
1	0,0001	144	20	0,77	Kategorizelenmiş Çapraz Entropi
2	0,0001	144	25	0,77	Kategorizelenmiş Çapraz Entropi
3	0,0001	144	20	0,67	Mutlak Ortalama Hatası
4	0,0010	144	20	0,74	Kategorizelenmiş Çapraz Entropi
5	0,0001	96	25	0,85	Kategorizelenmiş Çapraz Entropi
6	0,0001	96	35	0,80	Kategorizelenmiş Çapraz Entropi
7	0,0001	128	35	0,80	Kategorizelenmiş Çapraz Entropi
8	0,0001	128	50	0,88	Kategorizelenmiş Çapraz Entropi

9	0,0001	128	100	0,92	Kategorizelenmiş Çapraz Entropi
10	0,0001	128	200	0,95	Kategorizelenmiş Çapraz Entropi

Çizelge 3.2'den takip edileceği üzere aynı veri seti [38] üzerinde PCA tekniği kullanarak ve MLP mimarisinin katman sayılarındaki düğüm sayıları soldan sağa: 28, 20, 20 ve 4 olacak şekilde belirlenmiştir. Öncellikle farklı kayıp fonksiyonları denenmiştir fakat mutlak ortalama hatasının (İng. absolute mean error) öğrenme sonuçları, kategorizelenmiş çapraz entropi (İng. categorical cross entropy) kayıp fonksiyonun çok gerisinde gelmiştir. Sonrasında öğrenme oranı artırılmıştır fakat bu durumda da düğüm ağırlıklarının güncelleme aralıkları artmıştır [48]. Sonrasında diğer hiper parametreler sabit tutulup toplu boyut değiştirilmiştir. En son denemelerde öğrenme süresi uzun sürse de %90 üzerinde doğruluk değerlerine ulaşılabilmiştir. Fakat öğrenme eğrilerinde birkaç iterasyonda çok keskin geçişler yaşanmıştır.



Şekil 3.6. Çizelge 3.2'te belirtilen 7. (a, b) ve 9. (c, d) sıradaki işlemlere ait öğrenme eğrileri.

Şekil 3.6'daki (a) ve (b) öğrenme eğrileri Çizelge 3.2'de 7. sıradaki işleme ait iken (c) ve (d) ise 9. sıradaki işleme aittir. Şekil 3.6'daki (a) ve (b) öğrenme eğrisi ile (c) ve (d) öğrenme eğrisi arasındaki tek farklı değişken iterasyon miktarıdır. Tasarım akışı takip edilerek, öğrenme süreçlerinde gelişim elde edilmiştir. Fakat Şekil 3.6'daki (c) öğrenme eğrisi yüksek doğruluk değerine ulaşmasına rağmen pürüzsüz değildir. Ek olarak bu öğrenme eğrisinin bazı zamanlarında doğruluk değerinde düşüşler görülmektedir. Bu düşüş değeri, (d) öğrenme eğrisindeki kayıp değerin artmasıyla da takip edilebilir. Şekil 3.6'daki öğrenme eğrisinde görülen bu çıkıntılar toplu boyutun büyük olduğu zamanda görünürler. Çünkü toplu boyut büyüdükçe modelin genelleme yapabilme yeteneği azalır. Bu çıkıntılar öğrenme parametresinin dağılımında bozulmalara neden olur [49]. Fakat toplu boyut çok küçük seçilirse de öğrenme süresi artar ve eksik öğrenme ile de karşılaşılabilir. Bu çıkıntılardan kurtulmak için öğrenme değeri düşürülmüştür ve kayıp fonksiyonu değiştirilmiştir. Bu güncellemelere ek olarak adaptif öğrenme fonksiyonu eklenerek eğri üzerinde beklenmedik iniş ve çıkışlar giderilmiştir [49]. En son elde edilen Şekil 3.6'daki (c) ve (d) olarak isimlendirilen öğrenme eğrilerinde halen çıkıntılar oluşmuştur. Bu çıkıntılar giderilemediğinden dolayı bu durumun PCA yönteminin kullanılmasından kaynaklı olduğu düşüncesine varılmıştır. Bundan dolayı PCA yerine alt örnekleme yöntemi kullanarak iteratif öğrenme sürecine devam edilmiştir.

Çizelge 3.3. Seçilen veri seti [38] kullanılarak alt-örnekleme ile elde edilen sonuçlar.

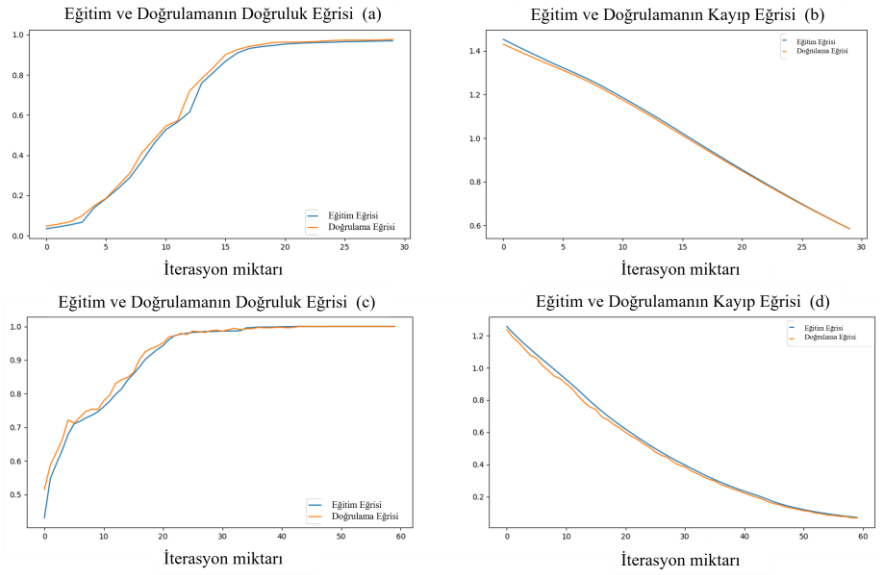
İşlem Sırası	**Öğrenme Oranı**	**Toplu Boyut**	**İterasyon**	**Veri Setinin Eğitim / Doğrulama**		**F1 Puanı**
				Doğruluk Oranı	**Kayıp Oranı**	
1	0,0001	128	30	0,9691 / 0,9764	0,5851 / 0,5385	0,7000
2	0,0001	128	40	0,9931 / 0,9958	0,2260 / 0,2153	0,8506
3	0,0001	128	50	0,9976 / 0,9972	0,1171 / 0,1101	0,8271
4	0,0001	144	25	0,9542 / 0,9681	0,6953 / 0,6753	0,8281
5	0,0001	144	30	0,9208 / 0,9111	0,5680 / 0,5775	0,9046
6	0,0001	144	40	0,9868 / 0,9889	0,5408 / 0,5359	0,7750
7	0,0001	144	40	0,9997 / 0,9986	0,2118 / 0,2108	0,8089

8	0,0001	144	60	1,000 / 1,000	0,0719 / 0,6882	0,9237
9	0,00005	144	45	0,9635 / 0,9597	0,6781 / 0,6637	0,7557
10	0,00005	144	60	0,9913 / 0,9944	0,4467 / 0,4423	0,9348

Çizelge 3.3'te kullanılan MLP modelinde girdi katmandan, çıktı katmana kadar sırasıyla: 36, 20, 20 ve 4 düğüm kullanılmıştır. Çizelge 3.2'den takip edilebileceği üzere sadece eğitim ve doğruluk oranı incelenerek AI modelinin oluşturulması yanıltıcı olabilir. Bundan dolayı Çizelge 3.3'te F1 puanı da eklenmiştir. F1 puanı (diğer ismiyle F-puanı), daha yüksek duyarlılığa (İng. recall) sahip algoritmalara fayda sağlayan ve daha yüksek kesinliğe (İng. precision) sahip algoritmalara karşı modelin başarısını gösteren bir bileşik ölçüdür [50]. F1 puanı formülü, AI modelinde duyarlılık ve kesinlik arasındaki ilişki kullanılarak Denklem 3.3 ile ifade elde edilir [50]:

$$F1\ puanı = \frac{(\beta^2+1) \times kesinlik \times duyarlılık}{\beta^2 \times kesinlik + duyarlılık} \tag{3.3}$$

Çizelge 3.3 incelendiğinde yüksek F1 puanı ve yüksek doğruluk elde etmek için hiper parametreler güncellenmiştir. Bu süreçte Şekil 3.3'teki iteratif öğrenme süreci takip edilmiştir. Tasarlanan ve belirlenen hiper parametrelere göre seçilen veri seti [38] için yüksek doğruluk ve yüksek F1 puanına ulaşılabilmiştir.

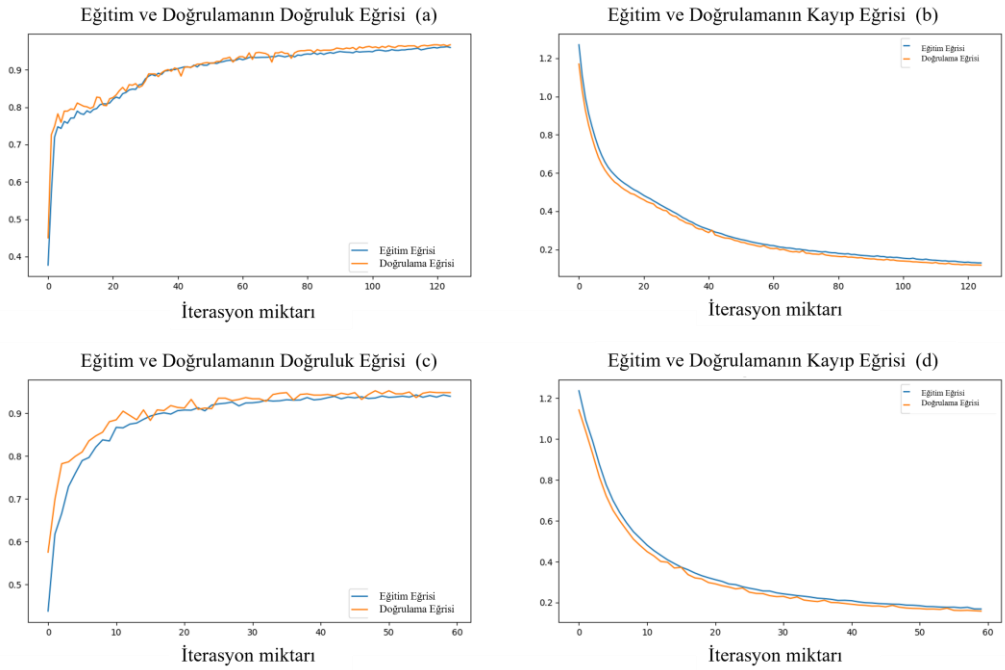


Şekil 3.7. Çizelge 3.3'te belirtilen 1. (a, b) ve 8. (c, d) sıradaki işlemlere ait öğrenme eğrileri.

Şekil 3.7'de görüldüğü üzere kullanılan fonksiyonlar ve iteratif güncellenen hiper parametreler ile tasarlanan MLP modeli daha yüksek doğruluk, daha yüksek F1 puanı ve daha pürüzsüz öğrenme eğrilerine ulaşmıştır, ancak AI modelinin olgunlaşması için yeterli seviyede değildir. Bu çalışmalarda 5 farklı hazır veri seti kullanılmıştır [38-42]. 5 farklı hazır veri setlerine ait en iyi sonuçlar için Çizelge 3.4 incelenebilir.

Çizelge 3.4. Tasarlanan MLP modelinin farklı veri setlerinden elde edilen sonuçlar.

Veri Seti	**F1 Puanı**	**Duyarlılık**	**Kesinlik**	**Veri Setinin Eğitim / Doğruluk Oranı**
[38]	%93,5	%99,7	%88,0	%99,1 / %99,4
[39]	%62,1	%83,1	%49,7	%68,6 / %70,4
[40]	%82,2	%95,1	%72,4	%95,6 / %94,7
[41]	%79,7	%99,4	%66,6	%96,0 / %96,7
[42]	%90,2	%95,8	%85,2	%93,5 / %95,2



Şekil 3.8. Çizelge 3.4.'te belirtilen [41] (a ve b) ile [42] (c ve d) veri setlerine ait öğrenme eğrileri.

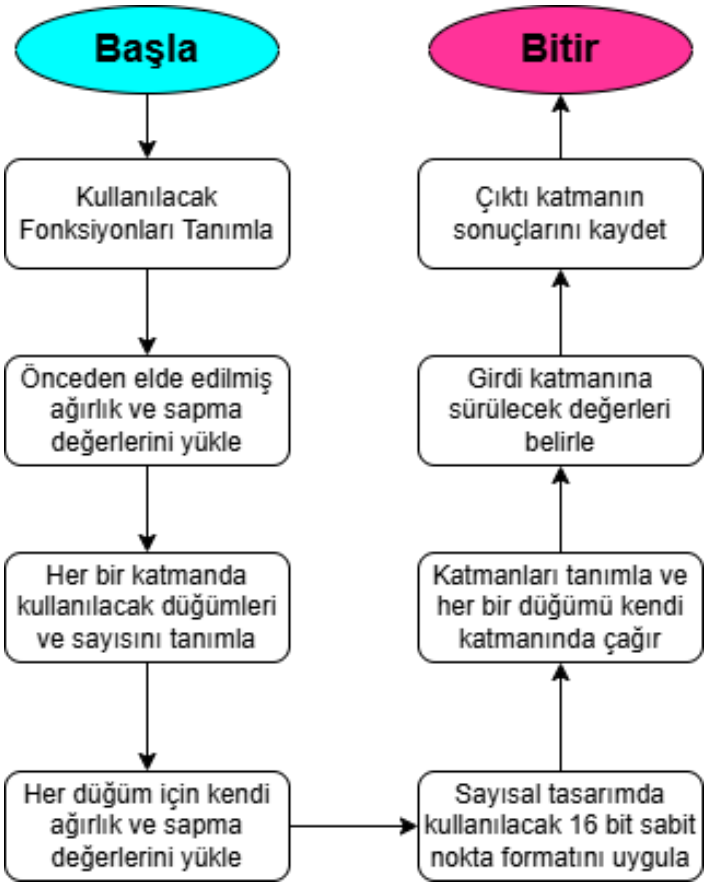
Şekil 3.8'de bulunan (a) ve (b) ile (c) ve (d) öğrenme eğrileri sırasıyla, Çizelge 3.4'te yer alan [41] ve [42] numaralı referanslara aittir. Çizelge 3.4'ün oluşturulma süresi boyunca yapılan geliştirmeler ile Çizelge 3.1'de bulunan değişkenlerin değerleri belirlenmiştir.

AI modeli oluşturulurken çıktı katmanındaki aktivasyon fonksiyonu tez çalışmasına özgü tercih edilmiştir. Aktivasyon fonksiyonlarından softmax fonksiyonu çok etiketli sınıflandırmalarda, sigmoid fonksiyonu ise ikili sınıflandırma problemlerinde daha etkilidir [23,24]. Tez çalışmasının birden çok sınıf arasından doğru sınıfı tahmin etme hedefinden dolayı, MLP mimarisinin çıktı katmanında aktivasyon fonksiyonu olarak softmax tercih edilmiştir. Gizli katmanlarda ise aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Şekil 3.8'den gözlemlenebileceği üzere veri seti miktarı, öğrenme sürecini etkilemektedir. Bu görselde kullanılan veri setinde [38] daha fazla örnek bulunduğu için daha kararlı bir öğrenme eğrisi elde edilmişken, Şekil 3.7'de kullanılan veri setlerinde [41,42] daha az örnek verisi bulunduğu için nispeten daha hızlı doğruluk değerine yakınsanmıştır, ancak öğrenme eğrileri daha kararsız hale gelmiştir.

Deneysel çalışmalar sonucunda CPS dizisi ile oluşturulacak veri setinin boyutunun tez çalışmasını etkileyeceği öngörülmüştür. AI uygulamalarında elde edilen sonuçlar doğrultusunda, oluşturulacak veri seti ile uygun doğruluk değerlerine ulaşılabileceği düşüncesine ulaşılmıştır [51]. Deneysel çalışmalar, uygun doğruluk, kayıp ve F1 puanı elde edene kadar uygulanmaya devam edilmiştir [51]. Deneysel çalışmalar sonucunda, Çizelge 3.3'teki 10. işlemin ağırlık ve sapma değerleri, MLP mimarisinin ön tasarımında kullanılmak için kaydedilmiştir.

3.2.1.2 Veri İşleme Sistemi için Yüksek Seviyeli Dil ile Sinir Ağı Tasarımı

Daha önce yapılan açıklamaların paralelinde Şekil 3.2'deki akış süreci takip edildiğinde sırada yüksek seviyeli dil ile sinir ağları tasarımı bulunmaktadır. Yüksek seviyeli dil ile sinir ağı tasarımının temel amacı, MLP mimarisi sayısal tasarıma aktarılmadan önce, oluşabilecek problemleri öngörmektir. Ek olarak MLP mimarisi kapı seviyesi tasarlandığında da önce sentez öncesi doğrulama ve yüksek seviyeli dil ile tasarlanan MLP mimari doğrulaması karşılaştırılmış olur. Yüksek seviyeli dil ile tasarlanan MLP mimarisinin süreç akış diyagramına Şekil 3.9'dan ulaşılabilir.



Şekil 3.9. Yüksek seviyede dil ile tasarlanan AI modelinin akış diyagramı.

Şekil 3.9'daki akış süreci için öncellikle yüksek seviyeli dil ile tasarım oluşturulur. Kaydedilen ağırlık ve sapma değerleri MLP mimarisine aktarılır. Ağırlık ve sapma değerleri yüklendikten sonra, Çizelge 3.1'de belirlenen katman ve düğüm değerleri tanımlanır. Yüklenen ağırlık ve sapma değerleri ilgili düğümlere aktarılır. Sayısal tasarımda sayı formatı olarak 16 bit sabit nokta tercih edilmiştir. Bu sayı formatı için ise 1 tane tam sayı ve 15 tane ondalık sayı formatı tercih edilmiştir. Çünkü ağırlık ve sapma değerleri incelendiğinde tam sayı değeri en fazla 1 veya -1 olmaktadır. Sonrasında her bir düğüm kendi katmanında, kendi ağırlık ve sapma değerleriyle tanımlanır. Böylece Şekil 3.9'daki akış diyagramı takip edildiğinde MLP mimarisi uygulanmaya hazır hale getirilmiş olur. Girdi katmanına ilgili değerler sürülerek çıktı değerleri gözlemlenir. Böylece sayısal tasarımda karşılaşılacak problemler öngörülebilmiştir.

Şekil 3.2'deki veri işleme sistemi akış şeması takip edildiğinde sırada MLP mimarisinin davranışsal modelini tasarlamak vardır. MLP mimarisi RTL seviyesinde, 16 bit sabit nokta sayı formatını kullanmıştır. RTL seviyesinde, 16 bit sabit nokta formatında

hassasiyeti artırmak için sayı formatı, Q1.15 yani 1 bit tam sayı ve 15 bit ondalık sayı olarak kullanılmıştır [52]. Kullanılan formata göre işlem sonuçlarının ondalık tabada karşılığının çıkarımı yapılmıştır.

$$2^{-N} = 10^{-N \log 2} \tag{3.4}$$

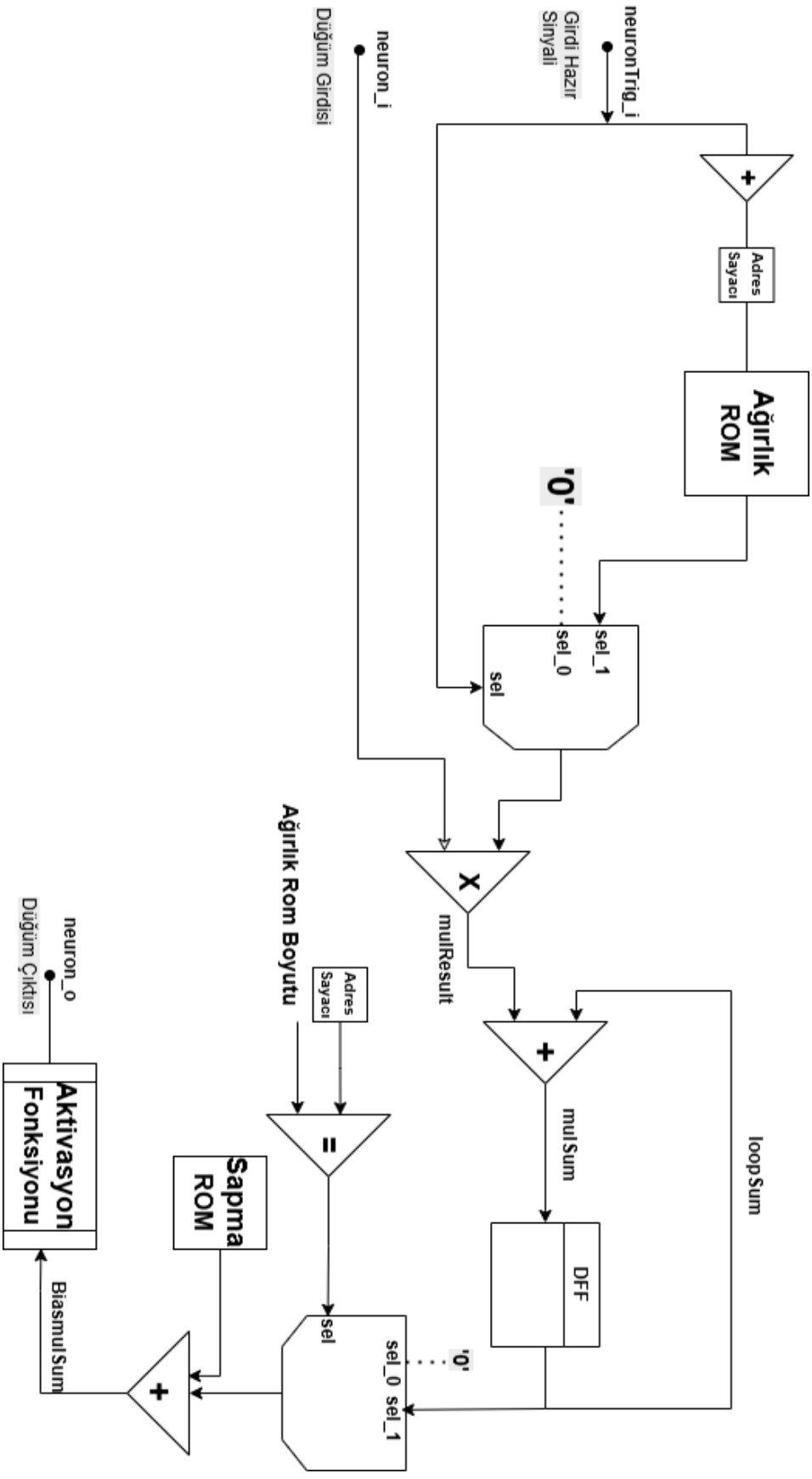
$$\#_{Ondalık\ Basamak} = N_{Sabit\ Nokta\ Kesir\ Biti} \times \log_{10} 2 \tag{3.5}$$

$$\#_{Ondalık\ Basamak} = 15 \times 0{,}3010 \approx 4{,}515 \tag{3.6}$$

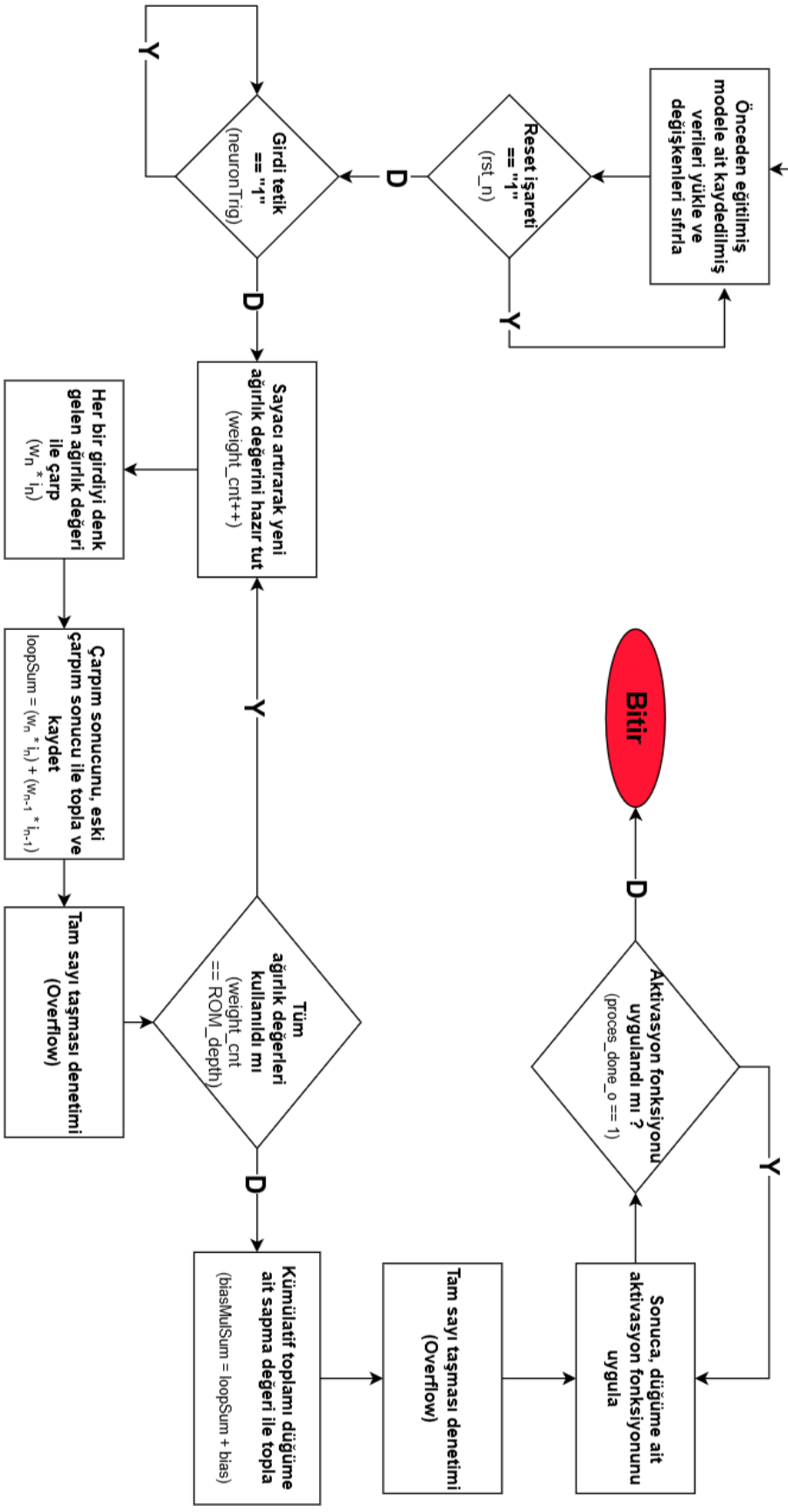
Denklemler 3.4 ve 3.5'de 16 bit sabit nokta formatının ondalık tabanda karşılığının hesaplanması için kullanılan formüller verilmiştir [52]. Denklem 3.6'ya göre kullanılan formatın ondalık kısmı N = 15 olmalıdır. Denklem 3.6 incelendiğinde, kullanılan 16 bit sabit nokta formatının, ondalık tabanda virgülden sonra kaç basamağa kadar ayırt etme gerçekleşebileceği görülebilir. Ondalık tabanda virgülden sonra 4 basamağa kadar hassasiyette çalışılmıştır. Literatürdeki çalışmalara göre softmax fonksiyonunun sigmoide benzer şekilde farklı sayısal tasarım yöntemleri bulunmaktadır [30,31]. Bu tasarım yöntemleri ve MLP mimarisinin kapı seviyesinde tasarımı 3.2.1.3. Veri işleme sistemi için sayısal tasarım başlığında anlatılmaktadır. Sayısal tasarım için sinir ağının davranışsal tasarımı için Şekil 3.9'dan takip edilebilir. Şekil 3.2'deki tasarım akışı takip edildiğinde, davranışsal tasarımdan sonra MLP mimarisinin kapı seviyesinde tasarımına geçilmiştir.

3.2.1.3. Veri İşleme Sistemi için Sayısal Tasarım

Şekil 3.10, MLP mimarisinde kullanılacak her bir düğümün kapı seviyesinde davranışını göstermektedir. Daha önceki bölümlerde FPGA ve ASIC için takip edilen tasarım akışı hakkında bilgilendirme yapılmıştır. Davranışsal tasarıma ait akış diyagramı Şekil 3.11'de verilmiştir.



Şekil 3.10. MLP mimarisinde düğümlerin kapı seviyesinde davranışsal tasarımı.



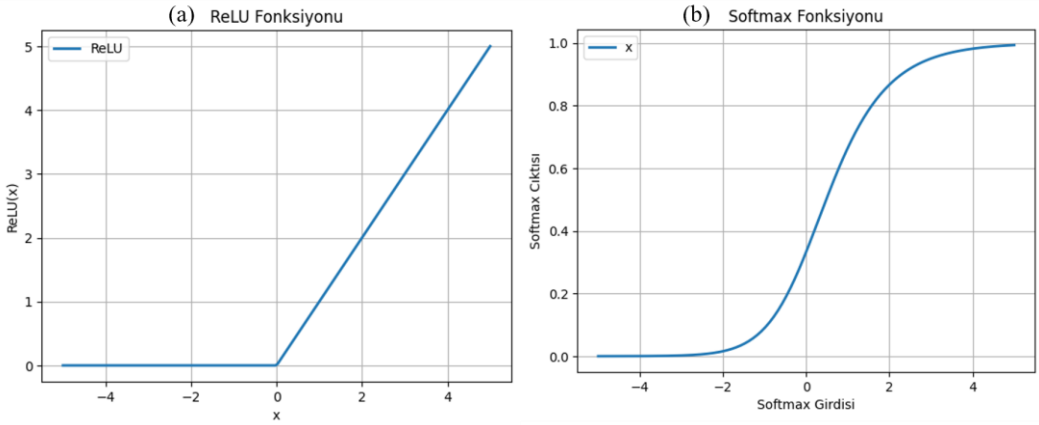
Şekil 3.11. MLP mimarisinde bulunan düğümlerin çalışma akış diyagramı.

Sayısal tasarıma başlanılmadan önce Şekil 3.10'ta bulunan davranışsal tasarım oluşturulmuştur. Şekil 3.11'de verilen akış diyagramı ise kapı seviyesinde tasarlanan düğümün çalışma sürecini göstermektedir. Davranışsal tasarımda bulunan işlemlerin denetim akışını planlamak için Şekil 3.11'de bulunan çalışma akış diyagramı tasarlanmıştır. Başlangıçta, önceden eğitilmiş AI modelinden gelen ağırlık ve sapma değerleri ilgili her düğüme sürülür. Bu değerler, düğüm içerisinde bulunan ROM'da (İng. Read Only Memory) saklanmaktadır. Bütün sistem tek sıfırlama tetiği tarafından denetlenir (*rst_n*). Tasarımdaki bütün sıfırlama tetikleri aktif düşük çalışmaktadır. Eğer bu tetik "0" değerine sahip ise bütün elemanlar ilk değerinde bekletilir. Sıfırlama tetiği "1"'e çekildikten sonra elemanlar işlem yapmaya hazır hale getirilir. Düğüm tasarımında sıfırlama tetiği saat ile senkron biçimde çalışabilir olması tercih edilmiştir. Çünkü hesaplama isteri yüksek olduğu için, sıfırlama tetik işaretinin saat sinyali ile örneklenip, kullanılması olası başlangıç değerine dönmeme durumu engellenmek istenmiştir. Sıfırlama tetik işareti kaldırıldıktan sonra girdi değerlerin denetimi, dışarıdan gelecek tetik girdi işareti ile sağlanır (*neuronTrig_i*). Tetik işareti, aktif yüksek geldiğinde sistemi tetiklemektedir. Bu sayede sadece girdi geldiği zaman düğüm aktif olacaktır Girdi tetik sinyalinin "1" olduğu her yükselen saat kenarı boyunca, ağırlık değerlerinin ROM'dan çekilmesini sağlayan sayaç artırılır (*weight_cnt*). Böylece sıradaki girdi değerinin kendisiyle eşleşmiş ağırlık değeriyle çarpılması sağlanır. Elde edilen her çarpım sonucu, bir değişkende saklanarak kümülatif toplama eklenir (*loopSum*). Ek olarak her toplama işlemi sonrasında tam sayı taşması (İng. overflow) denetimi yapılarak, matematiksel işlem sonuçlarında işaret kayması önlenir. Sonunda tüm ağırlık değerleri kullanılmıştır. Kümülatif toplam eğer var ise sapma değeri toplanır (*loopMulSum*). Böylece aktivasyon fonksiyonunun hesaplama aşamasına geçilmiş olur. Çıktı katmanındaki düğümlerde aktivasyon fonksiyonu olarak softmax fonksiyonu, diğer katmanlardaki düğümlerde ise ReLU aktivasyon fonksiyonu kullanılmaktadır. İlgili düğüm hangi aktivasyon fonksiyonunu kullanacak ise tam sayı taşıması denetiminden sonra toplam sonucunu alır ve işler. f_N değerine aktivasyon fonksiyonu uygulandıktan sonra düğüm, bütün işlemlerin tamamlandığını belirten bayrak işaretini dışarıya sürer (*process_done_o*). Böylece bir düğümün bütün çalışma akışı tamamlanmış olur.

Sayısal tasarım hem FPGA hem de ASIC için kullanacağından herhangi bir ticari ürüne bağlı kalmayacak biçimde kodlama yapılmıştır. Bunu sağlamak için ise olabildiğince yapısal bir tasarım oluşturulmuştur [34, 36]. Yapısal tasarım için her bir düğüm, bu

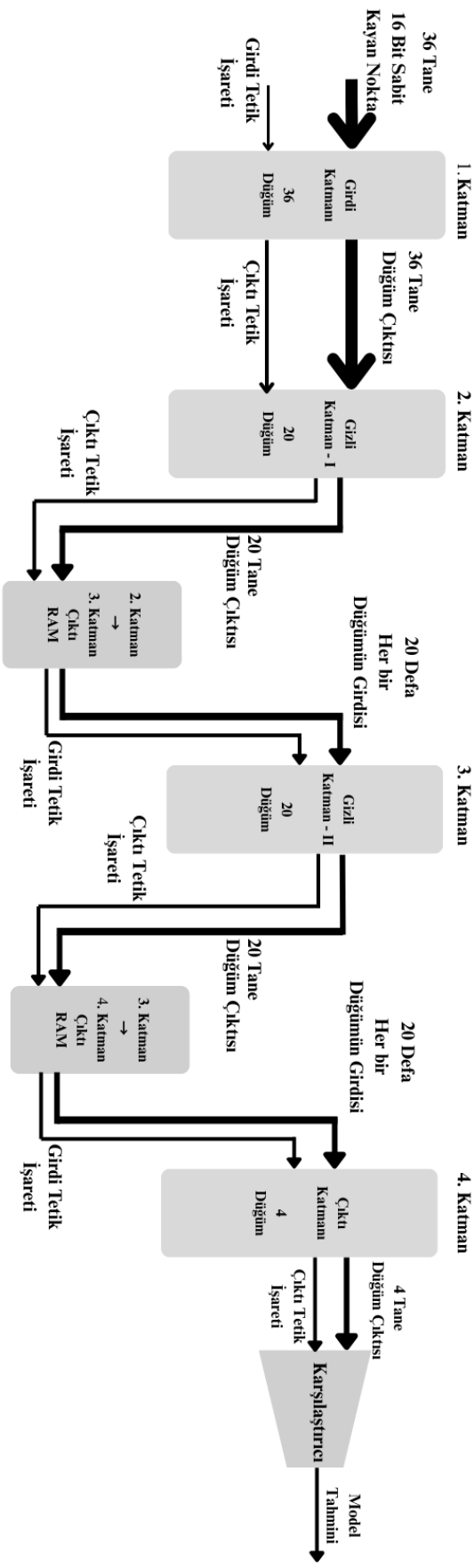
düğüme ait ağırlık değerlerinin saklandığı ROM ve sapma değerlerinin saklandığı ROM ayrı ayrı tasarlanmıştır.

ReLU aktivasyon fonksiyonu "0" değerinin altında bulunan girdileri eleyip, "0" değerinin yukarısında kalan girdilerin çıktıya sürülmesini sağlar [11]. ReLU aktivasyon fonksiyonu bu karakteristiğinden dolayı, sayısal tasarıma aktarılmaya uygundur. Fakat Softmax fonksiyonu doğrusal olmayan bir fonksiyondur [11].



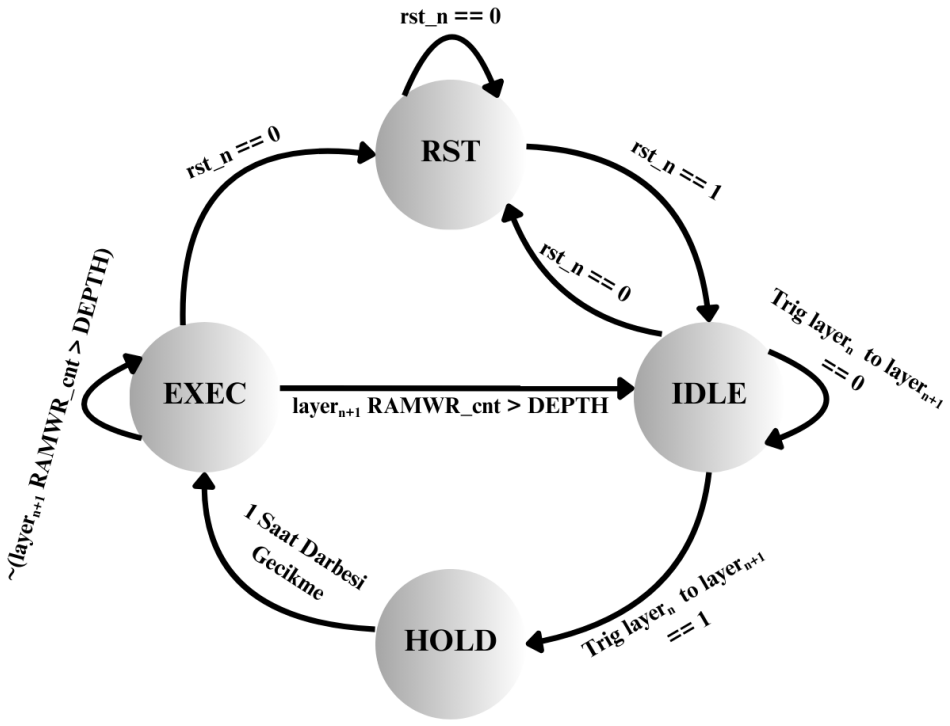
Şekil 3.12. Düğümlerde kullanılan aktivasyon fonksiyonları: a) ReLU fonksiyonu, b) Softmax Fonksiyonu.

Şekil 3.12'de ReLU (a) ve Softmax aktivasyon (b) fonksiyonlarının grafiklerine yer verilmiştir. Softmax ve Sigmoid fonksiyonu hem birbirine benzer hem de doğrusal olmayan fonksiyon oldukları için sayısal tasarımda gerçekleştirmek için literatür araştırmaları yapılmıştır [53-57]. Yapılan literatür araştırmaları sonucu softmax fonksiyonu, girdi değerine bağlı çıktı değerinin LUT veya bellek yapılarında saklandığı yöntem tercih edilmiştir [56]. Bu yöntem kullanılarak, sigmoid ile de yapıldığında yüksek doğruluk değerlerine ulaşılabilmiştir [57]. LUT tabanlı aktivasyon fonksiyonu tasarımı, parçalı fonksiyon biçiminde tasarıma göre daha yüksek hızlı sonuçlar üretecektir ve esnek yapılı tasarımı da desteklediği için tümleşik devrenin kaynak tüketimi daha düşük olmuştur [56]. Sayısal tasarımda, softmax fonksiyonunun değerleri -2 ve +2 arasındadır. Çünkü ağırlık ve girdi değerleri maksimum +1 veya -1 olabilir. Softmax fonksiyonunun sayısal tasarımını LUT'lar üzerinden gerçekleştirirken hesaplama hassasiyetini artırmak için -2 ve +2 arası toplam 1024 eşit parçaya bölünmüştür. Böylelikle küçük parçalara bölünen softmax fonksiyonu yüksek doğrulukta sonuç vermesi sağlanmıştır [57].



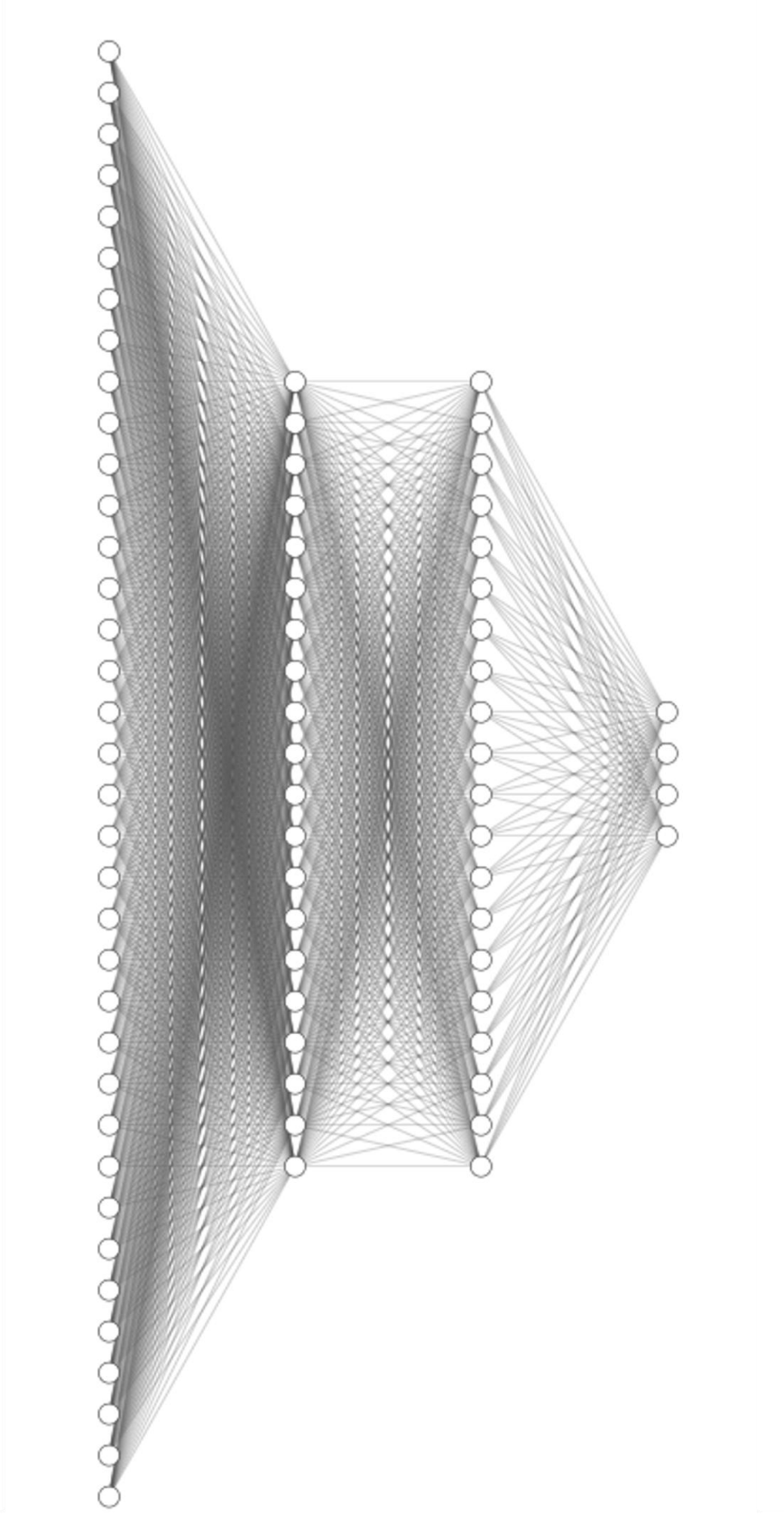
Şekil 3.13. MLP modeli sayısal tasarım blok diyagramı.

Şekil 3.13'te MLP modelinin sayısal tasarımına ait blok diyagramı verilmiştir. Ayrıca RTL sentezine ait blok diyagramı Şekil Ek.1'de verilmiştir. Girdi tetik işareti boyunca her biri sensörden gelen 16 bit sabit kayan nokta formatında veri paketi olmak kaydıyla 36 tane veri paketi MLP modeline sürülür. Sonrasında girdi katmanı yani 1. katmandaki her bir düğüm için Şekil 3.11'deki akış gerçekleştirir. İşlem bitti tetik işareti üretildiğinde 36 tane düğüm çıktısı yine aynı şekilde çıktı tetik işareti boyunca 36 saat darbesi boyunca sürülür. Katmanlar arasında veri aktarımında, veriler her seferinde yazmaçta (İng. register) saklanarak bütün sistemin senkronize biçimde denetimi sağlanmış olur. Şekil 3.13'te görüldüğü gibi 2. katmandan, 3. katmana geçişte; 3. katmandan, 4. katmana geçişte çıktı veriler RAM'de (İng. Read Access Memory) saklanır. Çünkü işlem yapılan katmada bulunan bütün düğümlerin, işlemlerinin tamamlandığından emin olunmalıdır. İşlem tamamlanması için düğüm işlemi tamamlandı tetiği denetlenir. Bundan dolayı hangi düğümün işlemi Şekil 3.11'deki akışta bitir bölümüne geldi ise çıktı değerleri RAM'de kendi numaralı adresine yazılır. Böylece sıradaki işlemde düğümlerin çıktıları sırayla, paralel olarak sıradaki katmanda bulunan bütün düğümlere senkronize biçimde sürülmüş olur. Katmanlar arası geçişte, düğüm çıktılarının denetimli biçimde aktarımı için sonlu durum makinesi tasarlanmıştır.



Şekil 3.14. MLP mimarisinde katman geçişleri denetimi için FSM tasarımı.

Şekil 3.14'te MLP mimarisinde düğümler arası veri iletim denetimini sağlayan sonlu durum makinesi (İng. Finite State Machine-FSM) görülmektedir. 2. katmanda 3. katmana ve 3. katmandan 4. katmana geçişte Şekil 3.14'te tasarlanan FSM kullanılmıştır. FSM, başlangıçta sıfırlama durumundan çıkmayı bekler (*rst_n*). Sıfırlama durumu bittikten sonra işlem yapılan katmanda bütün düğümlerin Şekil 3.11.'deki akışa uyarak, işlemlerinin sonlanmasını bekler (*Trig* $layer_n$*_to_*$layer_{n+1}$). Bu durum geçilmeden önce en az 1 saat darbesi beklenir. Bekleme süresi boyunca düğüm çıktıları, RAM'e yazılır. FSM, bu durumdan da çıktıktan sonra artık RAM'de saklanan veriler sıradaki katmanda bulunan düğümlere sürülmeye başlar. Düğümlere verilerin yazılabilmesi için tetik işareti gerekir. Bu tetik işareti de FSM tarafından sürülecektir. RAM aygıtındaki bütün veriler sürüldükten FSM, MLP modelindeki aynı akışı bütün işlemler bitene kadar devam ettirir. MLP mimarisi, son olarak çıktı katmana ait dört düğüm çıktısını karşılaştırıcıya sürmektedir. Karşılaştırıcı tasarımı, hangi düğüm çıktısı en büyük ise eğitimde kullanılan dört sınıftan ilgili sınıflandırmaya ait tetik işaretini "1" seviyesine çekerek MLP mimarisi tahminini gerçekleştirmiş olur. MLP mimarisi ve FSM kullanılarak bir yapay zeka yongası tasarlanmıştır. Şekil 3.2'deki akış takip edildiğinde, "yapay sinir ağını ve fonksiyonlarını mantık kapı seviyesi tasarla" aşaması da sağlanmış olmaktadır. Şekil 3.15'te kararlaştırılan MLP mimarisinin iç bağlantılarını gösteren tasarım, [58] kullanılarak çizilmiştir.

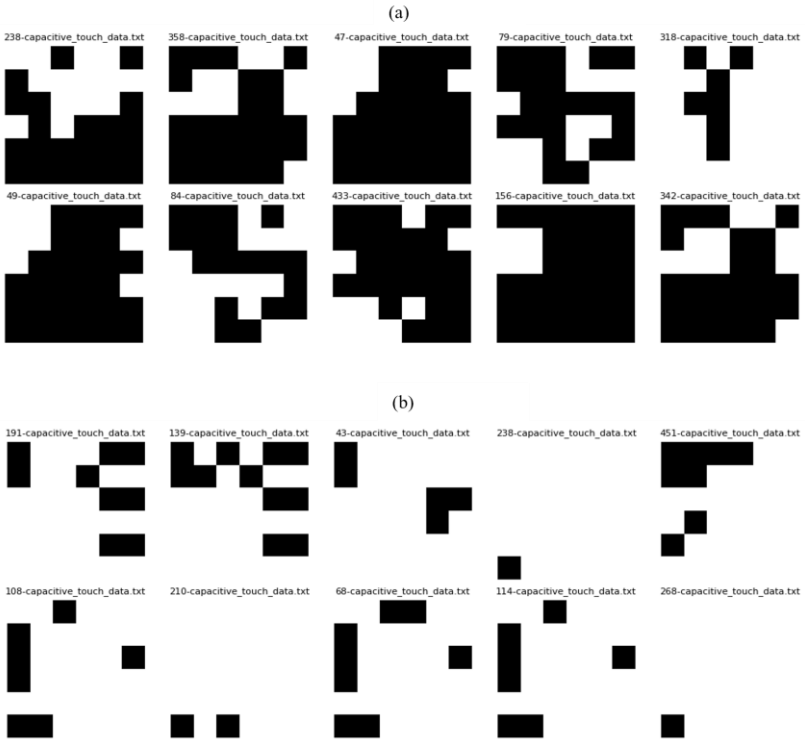


Şekil 3.15. MLP mimarisinin kararlaştırılmış halinin diyagramı.

4. OLUŞTURULAN VERİ SETİ ÜZERİNDEN ÇALIŞMALAR

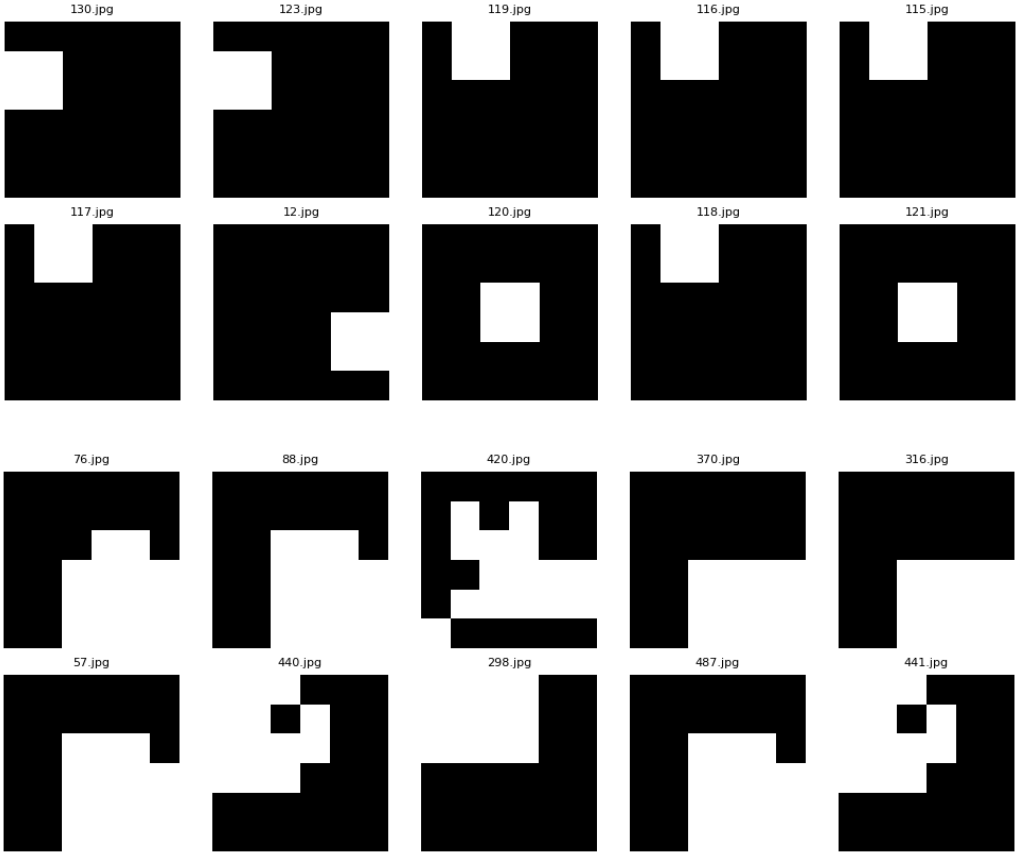
4.1 CPS Dizisinden Dokunma ile Alınan Verinin Analizi ve Çalışması

CPS dizisi 6x6 olacak şekilde toplam 36 sensörden oluşmaktadır. Başlangıçta CPS'den her bir sınıf için 500'er tane olacak şekilde toplam 2000 tane dokunma verisi toplanmıştır. Toplanan verilerin sınıflandırılması: 2 tane kareye dokunma, 4 tane kareye dokunma, 9 tane kareye dokunma ve 16 tane kareye dokunma şeklinde tercih edilmiştir. CPS dizisinden dokunma ile toplanan veriler incelendiğinde, dokunulan kare sayısı arttıkça etkileşime girilmeyen dizilerde görülen gürültünün büyüklüğü artmaktadır. Bu durumun veri toplanırken, aynı anda birden fazla CPS'ye dokunma sırasındaki hatalardan kaynaklanabileceği gibi, CPS dizilerinin de birbirinden etkilenmesinden dolayı olabileceği düşünülmektedir [5]. Safanova vd.'nin çalışmasından faydalanılarak, dokunma için oluşturulan veri setinin boyutunun AI modeli için küçük olduğu anlaşılmıştır [59].



Şekil 4.1. CPS dizilerindeki dokunma değerlerinin aynı anda 4 kareye dokunma (a) ve aynı anda 16 kareye dokunma (b) işlemlerinin ön işlem uygulanmadan görselleştirilmesi.

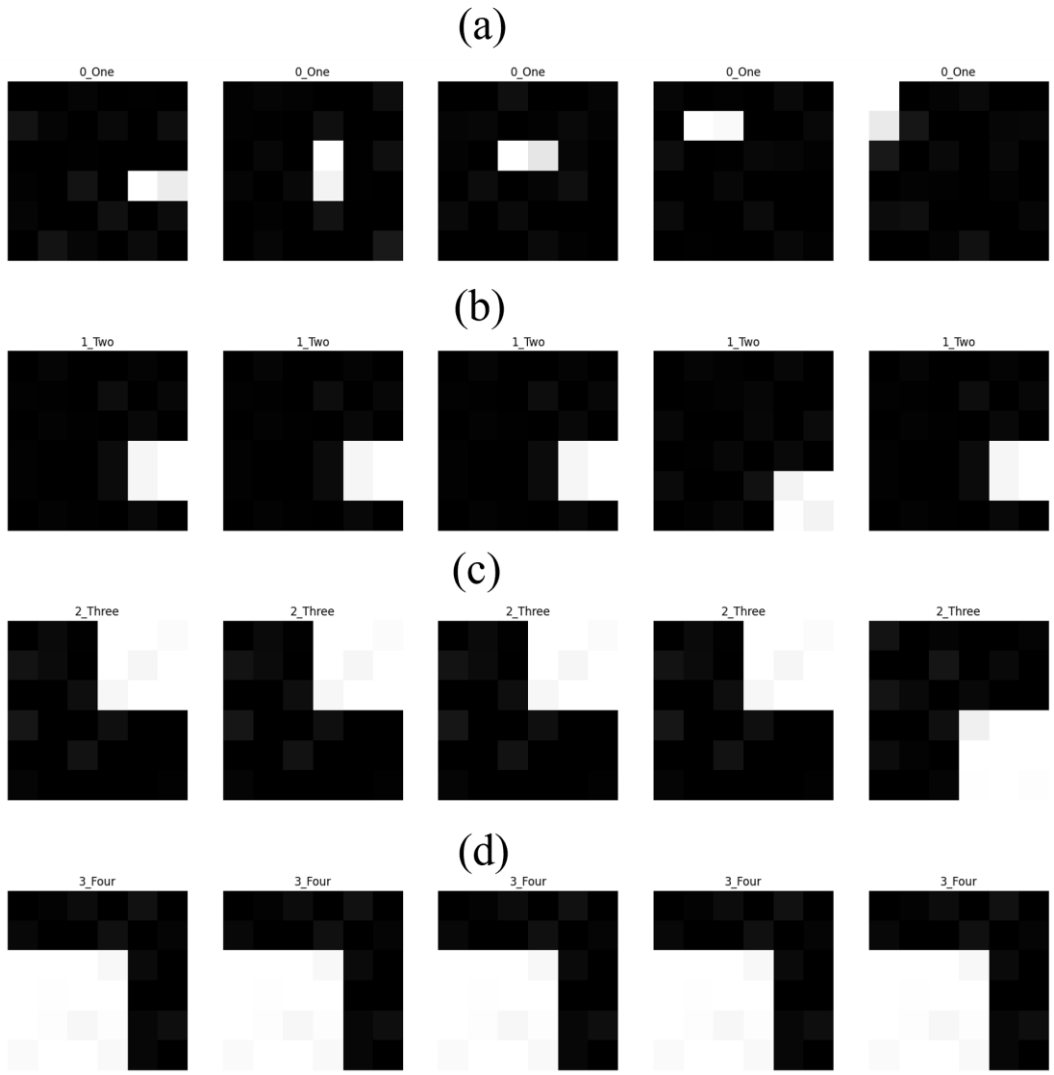
Şekil 4.1’de eğer ilgili dizinin sayısal değeri “0” ise siyah ve “0”’dan farklı ise beyaz renk ataması yapan görüntü işleme algoritmasına ait sonuçlar görülmektedir. Şekil 4.1(a) aynı anda 4 kareye dokunma ile Şekil 4.1(b) ise aynı anda 16 kareye dokunma ile elde edilmiştir. Şekil 4.1’de görüldüğü üzere ham data görselleştirilirken eğer eşik değeri belirlenmez ise CPS dizilerinin birbirlerine etkileşimi ayırt edilemez hale gelebilmektedir. İki farklı sınıfa ait veriler incelendiğinde her bir sınıf için farklı eşik değeri belirlenmiştir. Eşik değerleri okunan sayısal değerler analiz edilerek belirlenmiştir. Belirlenen eşik değerleri sonucunda ön işlemden geçirilen verilere ait yeni görseller Şekil 4.2’de sunulmuştur.



Şekil 4.2. CPS dizilerindeki dokunma değerlerinin aynı anda 4 kareye dokunma (a) ve aynı anda 16 kareye dokunma (b) işlemlerinin ön işlem uygulanarak görselleştirilmesi.

Şekil 4.2’de görüldüğü üzere CPS dizilerinden toplanan dokunma verilerine ön işlem uygulandığında veriler, daha ayırt edilebilir biçime gelmiştir. Ön işlem sonucu oluşturulan görseller AI modeli için etiketlendirilerek, eğitim için hazırlanmıştır. Ön

işlem uygulaması doğrulandıktan sonra, dokunmaya ait bütün veriler etiketlendirme için sınıflandırılmıştır. Aynı anda 2 kareye dokunma işlemi için “Sınıf 0”, aynı anda 4 kareye dokunma işlemi için “Sınıf 1”, aynı anda 9 kareye dokunma için “Sınıf 2” ve aynı anda 16 kareye dokunma için “Sınıf 4” etiketleri oluşturulmuştur. Şekil 4.3’te Sınıf 0 (a), Sınıf 1 (b), Sınıf 2(c) ve Sınıf 3(c)’e ait 5 rasgele görsele sunulmuştur.



Şekil 4.3. CPS dizilerine dokunarak elde edilen farklı sınıflara ait görseller a) Sınıf 0, b) Sınıf 1, c) Sınıf 2 ve d) Sınıf 3.

Şekil 4.3’te görüldüğü üzere sınıflandırmaya ait görsellerin birbirlerine benzediği ve ek olarak gri renkli piksellerin de oluştuğu görülmektedir. Çünkü CPS dizilerine dokunarak veri seti oluşturulurken, CPS’ler gürültüden etkilenmiştir. Gürültüden kaynaklı olarak AI

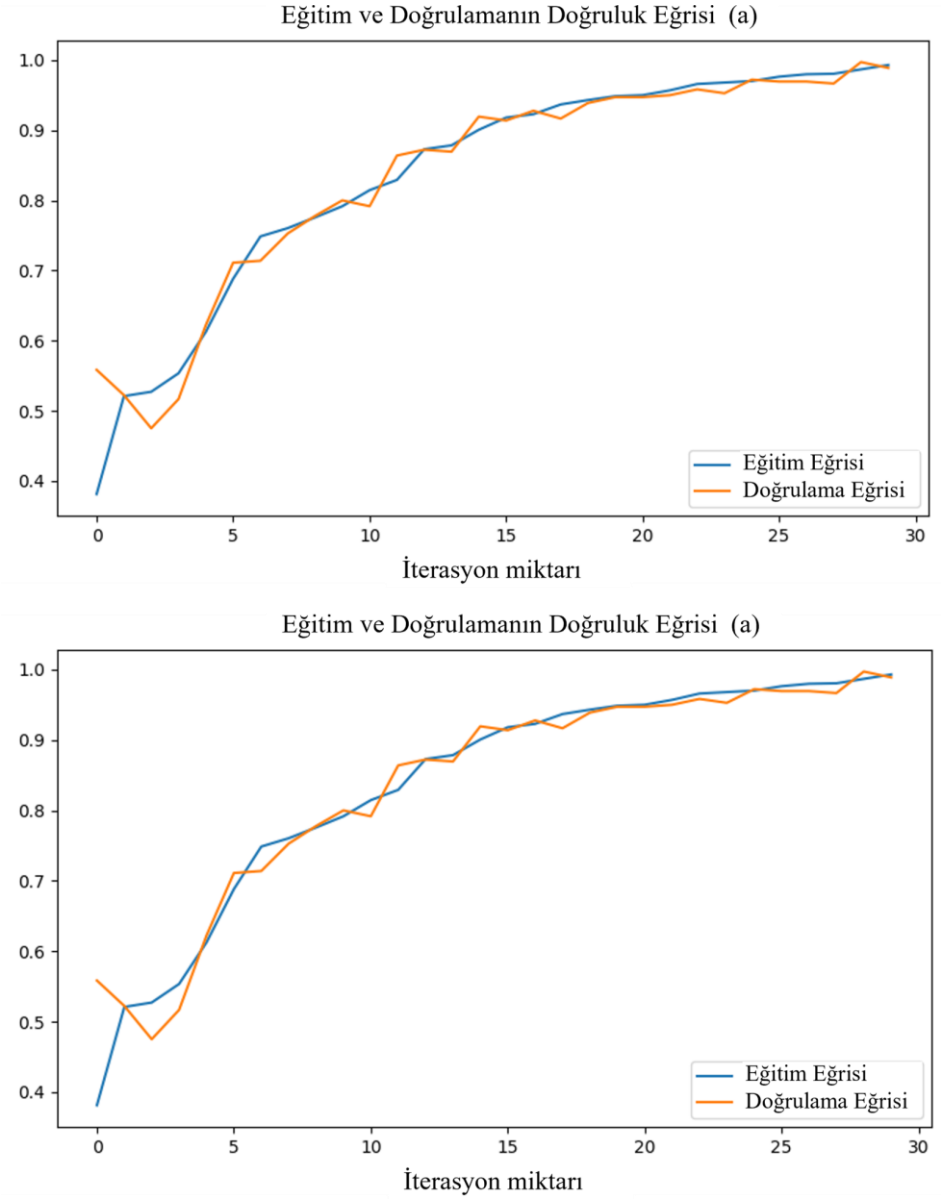
modelinin veri seti çeşitlenmiştir [61]. Oluşturulan veri seti 3.2.1.1. Başlığındaki çalışmalarda kullanılan AI modeli ile kullanılmıştır. Elde edilen sonuçlar, AI modele ait hiper parametrelerin güncellenmesiyle iyileştirilmiştir.

Çizelge 4.1. Oluşturulan veri seti ile sadece düzleştirme uygulanarak elde edilen sonuçlar.

İşlem Sırası	**Öğrenme Oranı**	**Toplu Boyut**	**İterasyon**	**Veri Setinin Eğitim / Doğrulama**		**F1 Puanı**
				Doğruluk Oranı	**Kayıp Oranı**	
1	0,00050	144	75	0,9736 / 0,9722	0,2260 / 0,2153	0,6557
2	0,00050	144	60	0,9757 / 0,9694	0,1171 / 0,1101	0,7708
3	0,00050	120	75	0,9910 / 0,9806	0,1171 / 0,1101	0,6685
4	0,00050	120	60	0,9924 / 0,9806	0,6953 / 0,6753	0,7385
5	0,00025	120	60	0,9125 / 0,9028	0,6953 / 0,6753	0,6360
6	0,00050	90	75	0,9917 / 0,9972	0,5408 / 0,5359	0,7400
7	0,00050	90	60	0,9944 / 0,9812	0,5408 / 0,5359	0,7484
8	0,00050	72	60	0,9972 / 0,9944	0,5408 / 0,5359	0,7479
9	0,00005	60	48	0,9967 / 0,9972	0,6781 / 0,6637	0,8820
10	0,00005	60	30	0,9972 / 0,9868	0,4467 / 0,4423	0,9207

Çizelge 4.1 incelendiğinde her ne kadar eğitim ve doğrulamaya ait doğruluk oranları birbirlerine yakın olsalar dahi her aşamada F1 puanının iyileşmesi gözlemlenebilir. [62]'deki yöntem incelenerek, F1 puanın iyileşmesi için toplu boyut miktarı her yeni işlemde azaltılmıştır. Böylelikle AI modelinin genelleştirme yapma kabiliyeti artırılmıştır. Öğrenme oranı düşürülme işlemi de denenmiştir. Fakat bu durumda da doğruluk oranlarının yakınsayamadığı (İng. converge) için daha yüksek iterasyon değerlerine ihtiyaç duyulmuştur. Doğruluk oranının bir değere yakınsaması için ise iterasyon miktarı artırılmıştır fakat bu durumda da öğrenme süreci uzamıştır. Ek olarak iterasyon miktarının yüksek olduğu uygulamalarda aşırı öğrenme riski artmaktadır ve AI modelinin de genelleme yapabilme kabiliyetinin azalacağı bilinmektedir [43]. Sonuç

olarak Çizelge 4.1'de işlem sırası takip edildiğinde en uygun sonuç işlem sıra numarası 10 olan sonuçtur. 10. işlem sırasına ait öğrenme eğrisi Şekil 4.4'te verilmiştir.



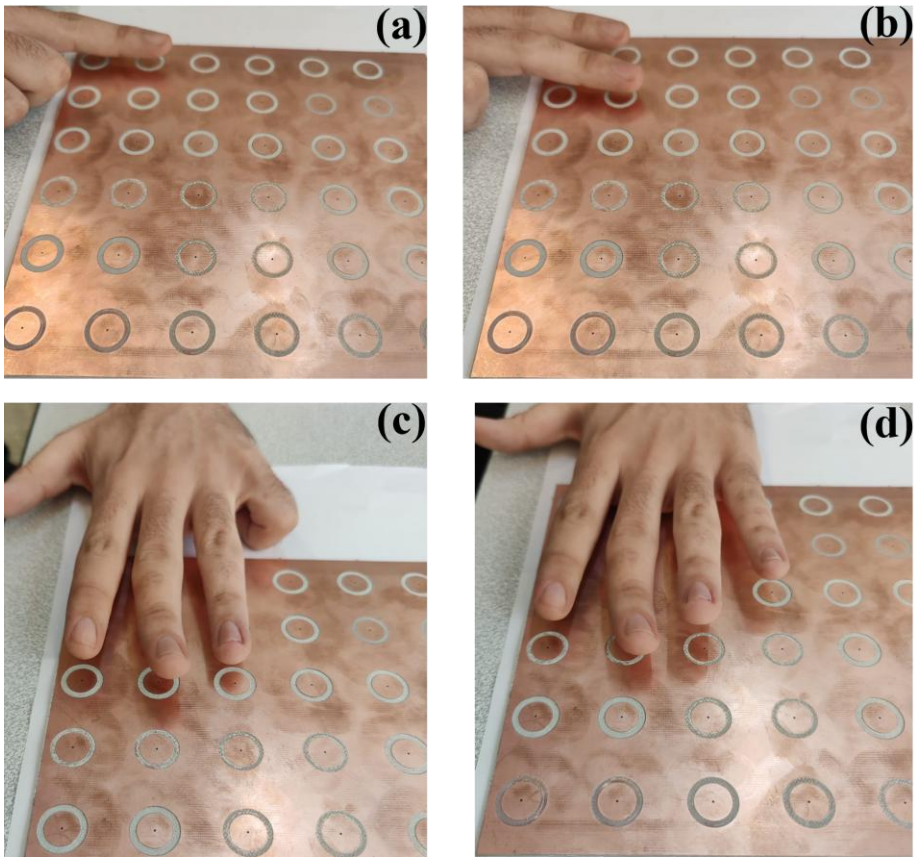
Şekil 4.4. Çizelge 4.1'de bulunan 10. işlem sırasına ait öğrenme eğrileri.

Şekil 4.4'te bulunan öğrenme eğrileri incelendiğinde kayıp fonksiyonuna ait eğrinin azalmaya devam ettiği ve eğitim ile doğrulamaya ait öğrenme eğrisinin doğruluk değerleri birbirleriyle benzer biçimde yükseldiği gözlemlenmektedir. Fakat Çizelge 4.1 incelendiğinde 10. işlemde doğruluk değeri yüksek eğitim verisinin, kayıp değerinin de daha yüksek olması aşırı ezberleme ihtimalini göstermektedir [62]. Bu durumun da eğitim

veri seti boyutunun düşük olmasından kaynaklanabilir [63]. Bundan dolayı Şekil 4.4'te bulunan öğrenme eğrisinin veya F1 skorunun incelenmesi yeterli olmamaktadır. Bunun için CPS dizilerine yaklaşılarak oluşturulan veri setinin miktarının daha fazla olması kararlaştırılmıştır. Ek olarak daha detaylı analiz için, oluşturulan veri setinin bir kısmı test verisi olarak ayrılmıştır.

4.2 CPS Dizilerine El Hareketlerinin Yaklaştırması ile Alınan Verinin Analizi ve Çalışması

Tezin ana çalışması, CPS dizilerine farklı el hareketleri yaklaştırılarak veri seti oluşturulmasıdır. Bu aşamada her bir sınıf için 3000'er tane olacak şekilde toplamda 12000 tane veri Hazar Haluk Aksaç tarafından toplanmıştır. Toplanan verilerin sınıflandırılması, 4.1. CPS Dizisinden Dokunma ile Alınan Verinin Analizi ve Çalışması başlığında bahsedilen sınıflandırma ile aynıdır. Fakat farklı olarak CPS dizilerine dokunma yerine el hareketi yaklaştırılarak veri seti oluşturulmuştur. Şekil 4.5'teki gibi farklı el hareketleri yaklaştırılarak her bir sınıf için veri seti oluşturulmuştur.



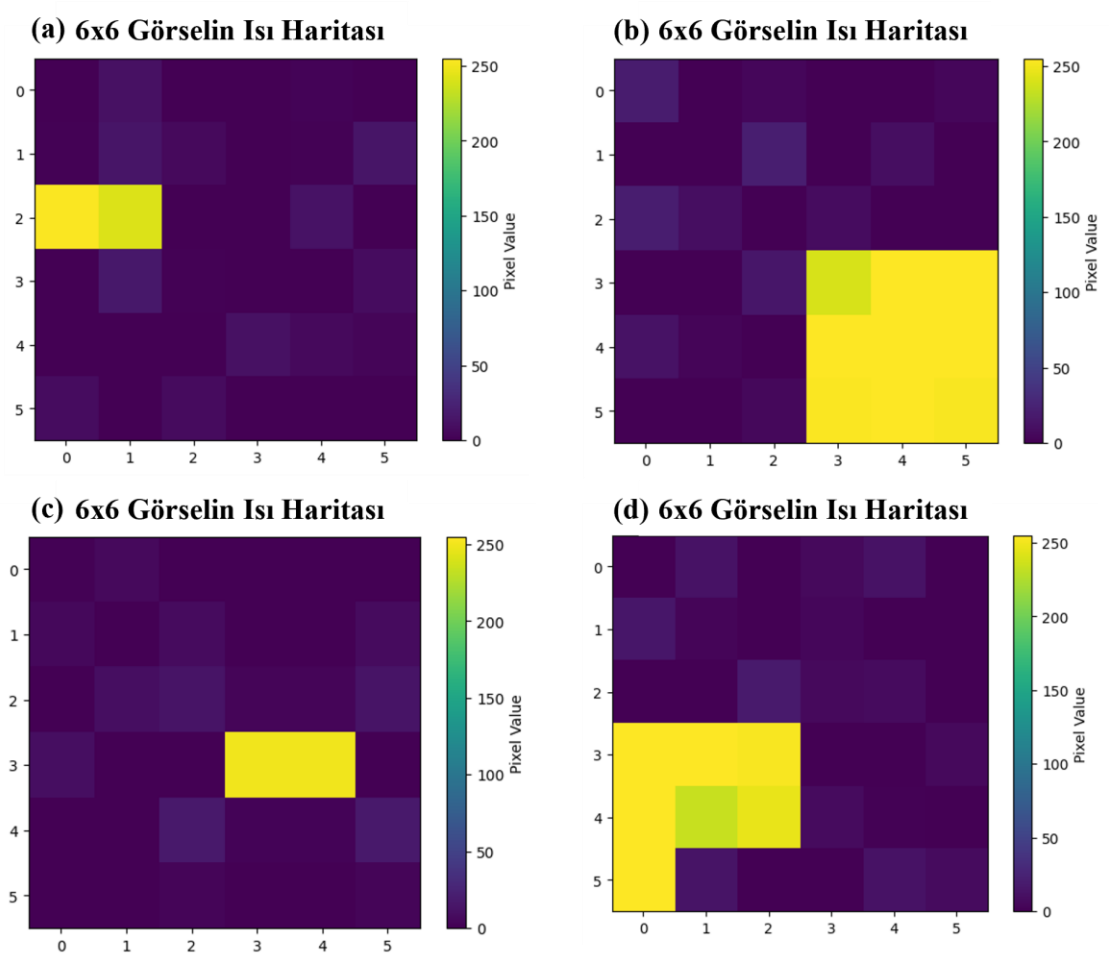
Şekil 4.5. CPS üzerine farklı el hareketleri yaklaştırılarak veri setinin oluşturulması

Şekil 4.5'te bulunan (a), (b), (c) ve (d) görselleri sırasıyla sınıf 0, sınıf 1, sınıf 2 ve sınıf 3 etiketlerine sahip veri setlerinin oluşturulma aşamalarını göstermektedir. Veri seti oluşturulurken el ile sensör arayüz paleti arasındaki mesafe 1 cm ile 2 cm arasındadır. Sensör arayüz paleti el bileğinden etkilenmeyecek şekilde veri toplanmıştır. Oluşturulan veri setinin detaylı analizi için bir kısmı test veri seti olarak ayrılmıştır. CPS dizilerine el hareketlerinin yaklaştırılması ile oluşturulan veri setinin yüzdelik dağılımı Çizelge 4.2'de verilmiştir.

Çizelge 4.2. CPS dizisine el hareketlerinin yaklaştırılması ile toplanan veri setinin eğitim için dağılımı.

İşlem	**Veri Seti Miktarı**	**Yüzdelik Karşılığı**
Eğitim	9000	%75
Doğrulama	1800	%15
Test	1200	%10

Çizelge 4.2'den görüldüğü üzere veri setinin dağılımı yapılırken en yüksek oran eğitim için ayrılmıştır. Çünkü AI modelinin hassasiyetini artırmak için eğitime ayrılan veri setinin yüzdelik dağılımı daha büyüktür [63, 64]. Deneysel çalışmalar ve literatür araştırmaları sonucunda [64, 65] Çizelge 4.2'de bulunan yüzdelik oranlar belirlenmiştir.

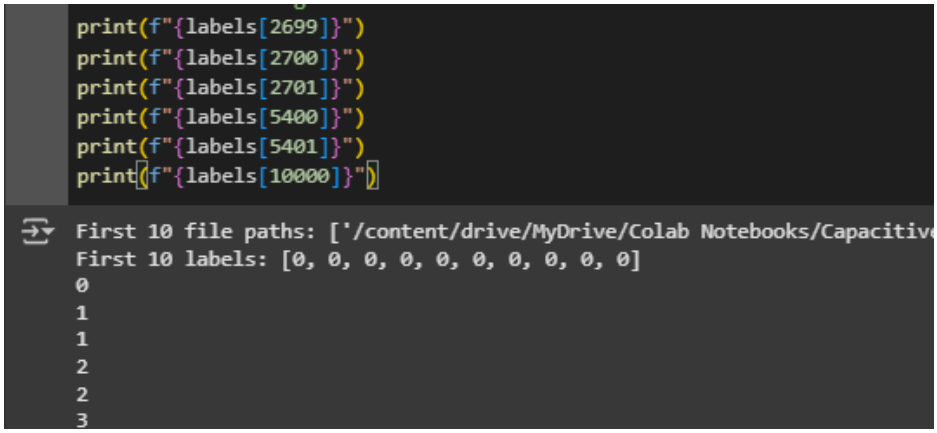


Şekil 4.6. CPS üzerine el hareketlerinin dokunarak ve yaklaştırılarak toplanan veri setlerinin örnek analizi: a) sınıf 0 dokunma, b) sınıf 2 dokunma, c) sınıf 0 yaklaşma ve d) sınıf 2 yaklaşma.

Şekil 4.6'da sınıf 0 (a,c) ve sınıf 2 (b,d) etiketlerine ait ısı haritaları görülmektedir. Şekil 4.6(a,b) ısı haritaları, dokunma ile elde edile veri seti iken, Şekil 4.6(c,d) ısı haritaları yaklaştırma ile elde edilen veri setleridir. Isı haritaları incelendiğinde rengin maviden sarıya doğru değişmesi ilgili CPS'lere daha çok yaklaşıldığını göstermektedir. Isı haritaları incelendiğinde, CPS dizilerinin hem birbirinden hem de ortamdan etkilendiği görülmektedir. Çünkü etkileşime girilmeyen CPS dizilerinde gürültü seviyesinde birtakım değerler görülmektedir. Isı haritalarında dokunma ile alınan verinin aksine yaklaşmadan kaynaklı, CPS dizilerinin birbirinden daha fazla etkilendiği görülmektedir. Hatta sınıf 2'ye ait dokunma ve yaklaşma ile elde edilen veri seti incelendiğinde, dokunma ile elde edilen veri seti tam algılanmış iken, yaklaşma ile elde edilen veri o an elin konumundan dolayı el hareketinin tam algılanamadığı görülmektedir. Bu durum toplanan veri setlerinin genelinde görülmektedir. Veri setinin boyutunun yeterli

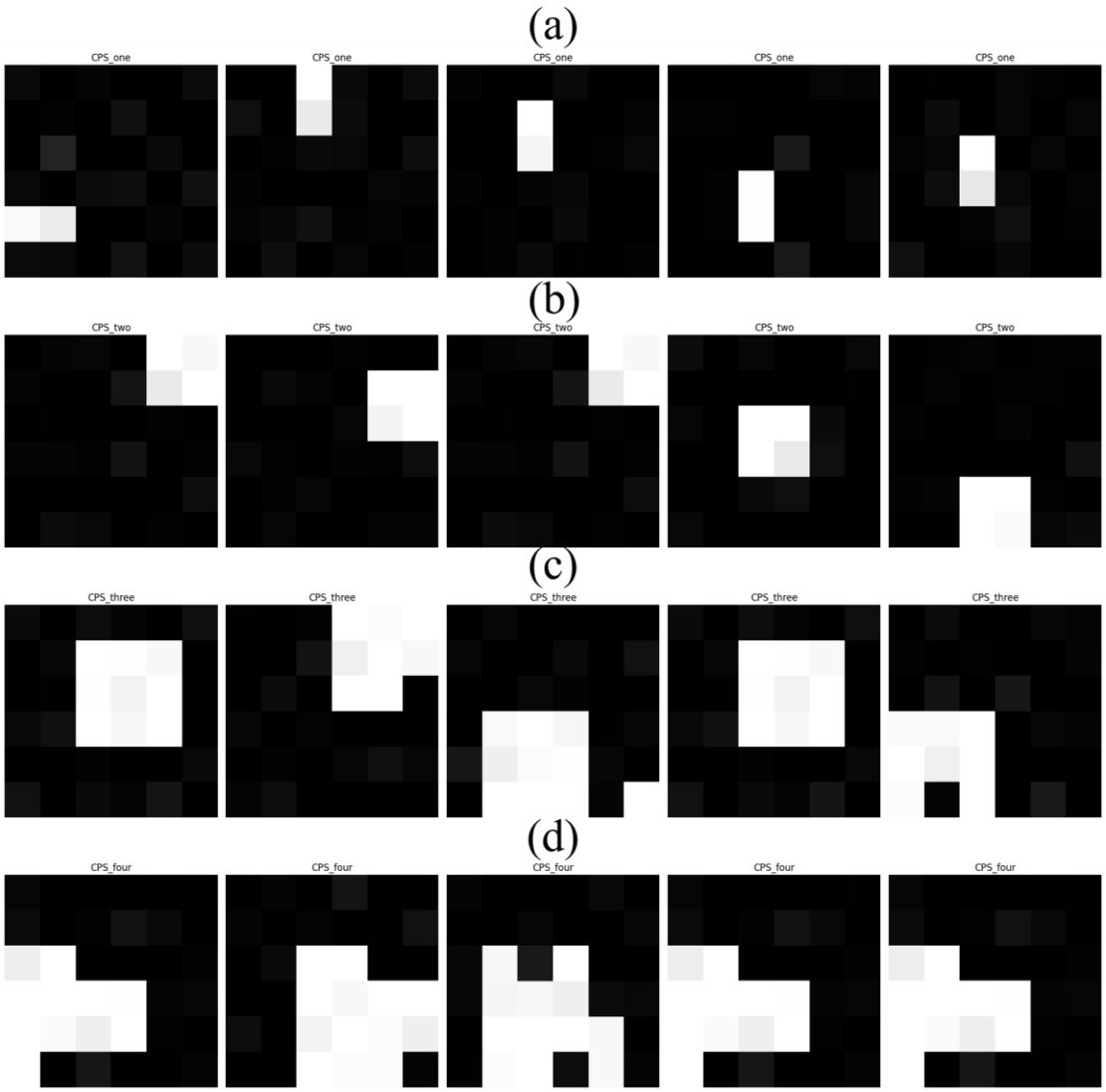
olmasından dolayı AI modelinin öğrenmesi için ek veri seti boyut artırımına ihtiyaç olmadığı düşünülmektedir [66]. Çünkü görüldüğü üzere gürültü miktarı sınıflandırmayı etkileyecek seviyede değildir. Ek olarak MLP mimarisinin toplamda 1244 eğitilebilir parametre içermesinden dolayı veri setinin de boyutunun yeterli olduğu anlaşılmaktadır [59, 60].

Çalışmalar kapsamında her bir sınıf için ayrı belirlenen eşik değerleri ile görselleştirme yapılmıştır. Farklı görüntü işleme yöntemleri kullanılarak ham veri setleri üzerinden analizler gerçekleştirilmiştir. Analizler sonucunda da gürültü değerlerini de sınıf veri setlerine dahil etmek için görüntüler gri ölçeklendirme ile tekrardan oluşturulmuştur. Gri ölçeklendirme ile elde edilen görseller öncellikle eğitim ve doğrulama veri setleri (her bir sınıf için 2700 veri) ve test veri seti (her bir sınıf için 300 veri) olacak şekilde ayrılmıştır. Eğitim ve doğrulama veri seti sonrasında Çizelge 4.2'deki yüzdelik dilimlere göre paylaşılmıştır. Böylelikle test veri seti, AI modelinin eğitimine dahil edilmeyerek, eğitimin gerçek performansının doğru değerlendirilmesine dikkat edilmiştir. Veri setinin etiketine göre sınıflandırmanın doğrulaması için Şekil 4.7'de etiket numarasına göre veri setinin hangi sınıfa dahil olduğu incelenmiştir. Şekil 4.7 incelendiğinde 2700 ve katları geçildikten sonra farklı etikete sahip görsellere ulaşılmaktadır. Böylelikle veri setinin her sınıfa eşit sayıda dağılım doğrulanması sağlanmıştır.



Şekil 4.7. Yaklaşma ile elde edilen verilerin sınıflandırılması ve doğrulanması.

Şekil 4.7'deki sınıflandırma doğrulamasından sonra CPS dizilerine el hareketlerinin yaklaştırması ile elde edilen görsellerin rasgele ön izlemesi görseller üzerinden sağlanmıştır. Ön izleme Şekil 4.8'de bulunan sınıflandırma üzerinden takip edilebilir.



Şekil 4.8. CPS dizilerine yaklaştırılan el hareketleri ile elde edilen görsellerin sınıflandırılması a) Sınıf 0, b) Sınıf 1, c) Sınıf 2 ve d) Sınıf 3.

CPS dizilerine el hareketi yaklaştırılarak oluşturan veri seti ile CPS dizilerine dokunarak oluşturulan veri seti aynı etiketlerden oluşmaktadır. (a) sınıfına ait etiket “Sınıf 0”, (b) sınıfına ait etiket “Sınıf 1”, (c) sınıfına ait etiket “Sınıf 2” ve (d) sınıfına ait etiket ise “Sınıf 3” olarak isimlendirilmiştir. Oluşturulan veri seti, 3.2.1.1. Başlığındaki çalışmalarda kullanılan AI modeli ile birlikte kullanılarak eğitim, doğrulama ve test sonuçları elde edilmiştir. Elde edilen sonuçlar, önceki çalışmalara benzer şekilde AI modele ait hiper parametrelerin güncellenmesiyle iyileştirilmiştir.

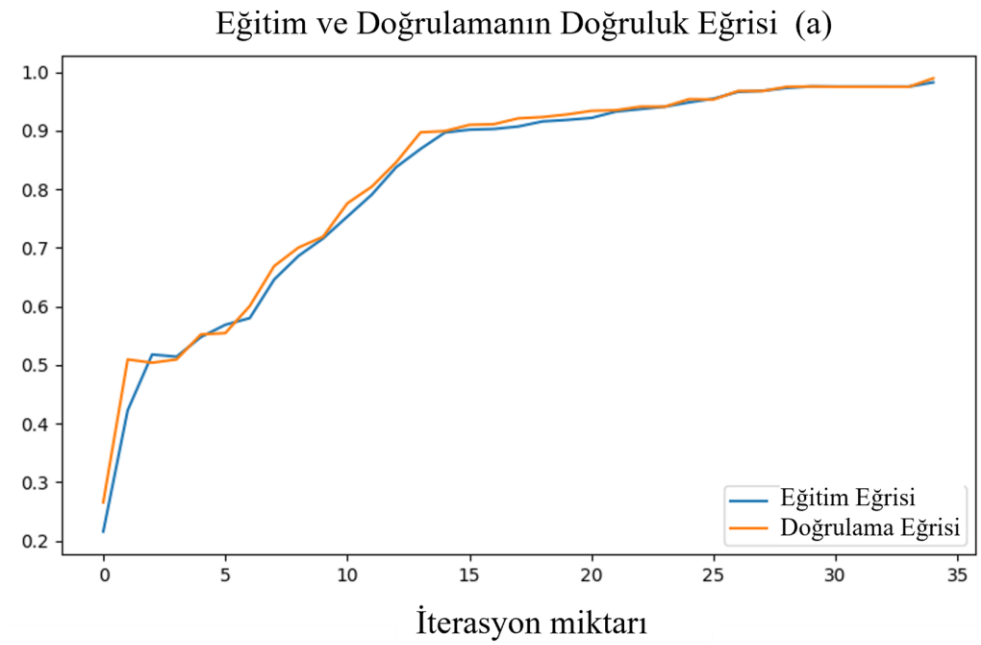
Çizelge 4.3. CPS dizilerine el hareketleri yaklaştırılarak oluşturulan veri setinin eğitim ve doğrulama sonuçları.

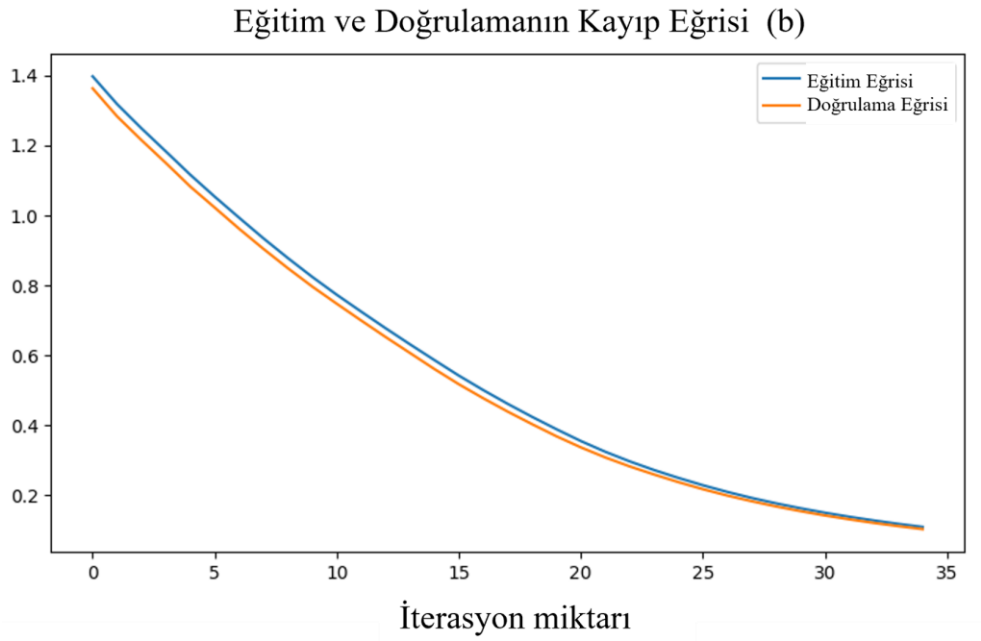
İşlem Sırası	**Öğrenme Oranı**	**Toplu Boyut**	**İterasyon**	**Veri Setinin Eğitim / Doğrulama**		**F1 Puanı**
				Doğruluk Oranı	**Kayıp Oranı**	
1	0,00010	60	45	0,9958 / 0,9950	0,0438 / 0,0421	0,9950
2	0,00010	60	30	0,9728 / 0,9678	0,1533 / 0,1526	0,9724
3	0,00010	48	25	0,9726 / 0,9672	0,1793 / 0,1767	0,9742
4	0,00010	48	30	0,9899 / 0,9839	0,0784 / 0,0834	0,9842
5	0,00010	48	35	0,9953 / 0,9956	0,0394 / 0,379	0,9956
6	0,00005	45	30	0,9314 / 0,9428	0,3098 / 0,2948	0,9349
7	0,00010	45	50	0,9958 / 0,9956	0,1207 / 0,0618	0,9239
8	0,00010	45	60	0,9976 / 0,9950	0,0157 / 0,0214	0,9950
9	0,00005	30	25	0,9696 / 0,9639	0,2516 / 0,2471	0,9475
10	0,00005	30	35	0,9826 / 0,9894	0,1101 / 0,1038	0,9820

Çizelge 4.4. CPS dizilerine farklı el hareketleri yaklaştırılarak oluşturulan veri setinin test sonuçları.

İşlem Sırası	**Test Doğruluk Oranı**	**Test Kayıp Oranı**	**F1 Puanı**
1	0,9950	0,0446	0,9125
2	0,9742	0,1453	0,8980
3	0,9817	0,1576	0,9027
4	0,9942	0,0702	0,9108
5	0,9908	0,1185	0,9117
6	0,9325	0,3720	0,8797
7	0,9958	0,1199	0,9125
8	0,9958	0,0166	0,9125
9	0.9700	0,2280	0.8970
10	0,9900	0,1013	0,9067

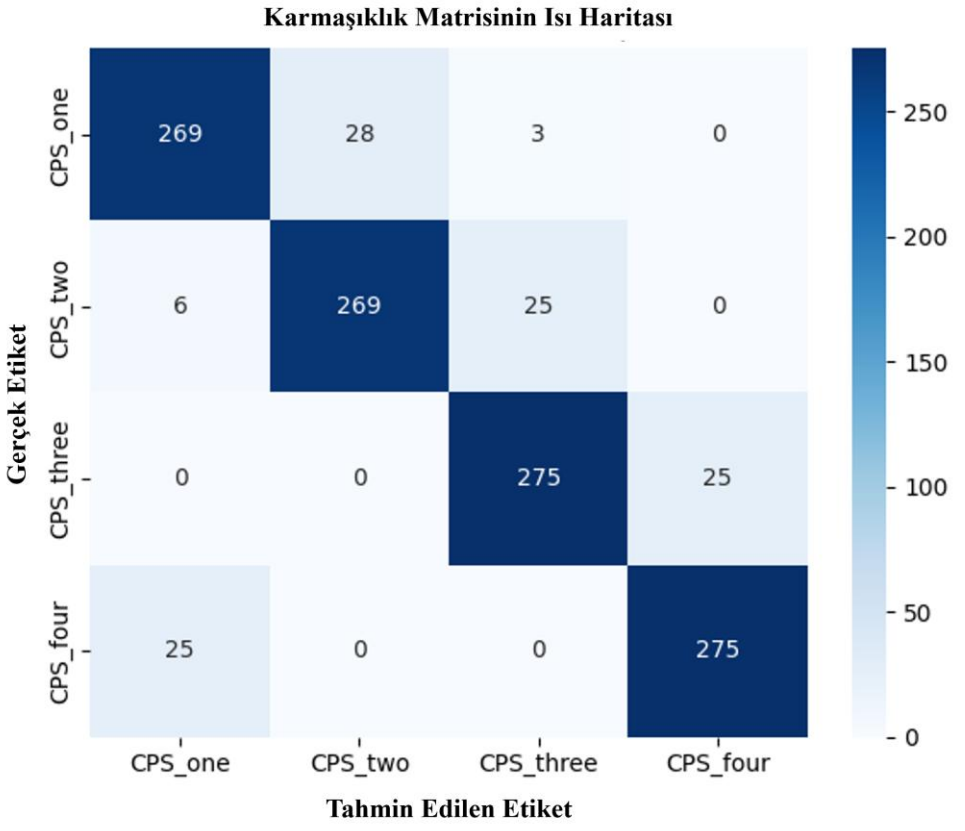
Çizelge 4.3'te eğitim sonuçlarına ait doğruluk ve kayıp oranı ile F1 puanları verilmiştir. Değerler incelendiğinde iteratif olarak her aşamada bütün değerlerin en uygunu elde edilmeye çalışmıştır. Çizelge 4.3 ve Çizelge 4.4 beraber incelendiğinde eğitim ve doğrulamaya ait değerlerin yüksek geldiği durumlarda test sonuçlarında halen F1 puanın düşük geldiği görülmektedir. Hem Çizelge 4.3 hem de Çizelge 4.4 sonuçları neticesinde hiper parametreler güncellenerek AI modelde iyileştirmeler yapılmıştır. Her iki çizelge için 10. işlemde de görüldüğü üzere bütün skorlar %90'dan yukarıda gelmektedir ve kayıp fonksiyonu değeri uygun seviyede gözlemlenmiştir.





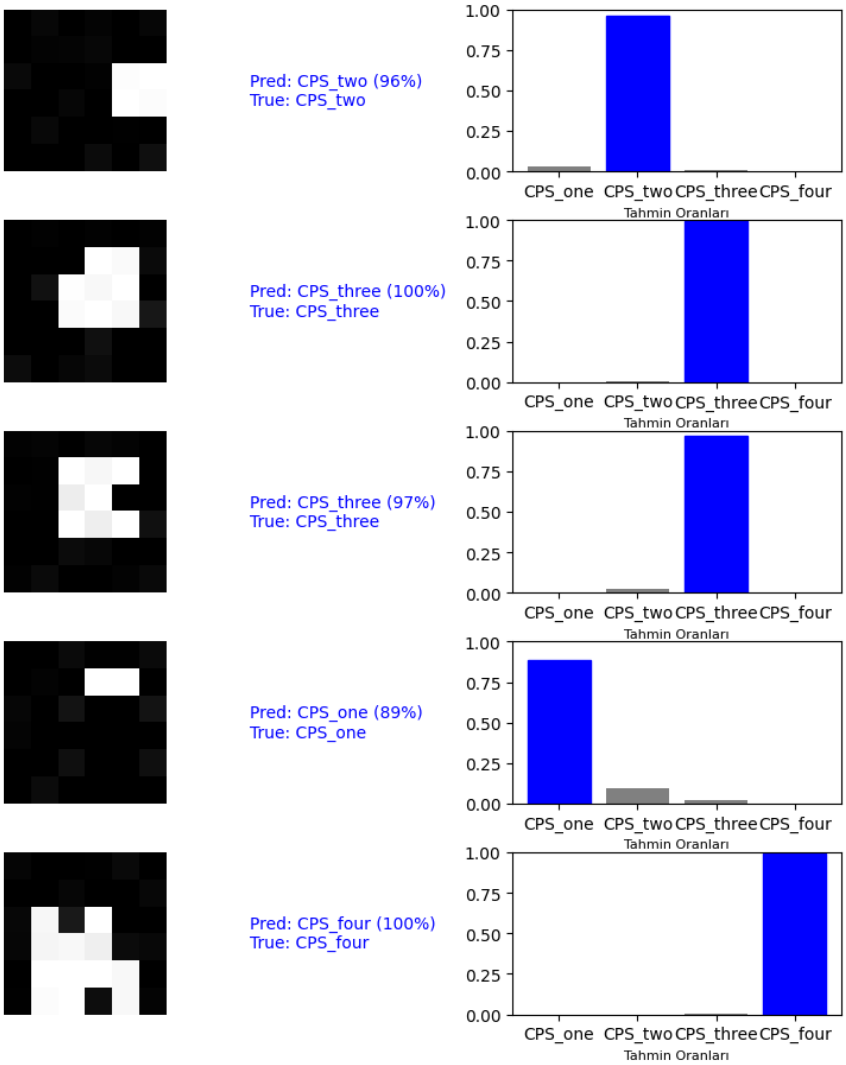
Şekil 4.9. El hareketi yakınlaştırmasıyla oluşturulan veri setine ait çizelge 4.3. ve 4.4'te bulunan 10. işleme ait öğrenme eğrisi.

Şekil 4.9'daki öğrenme eğrileri incelendiğinde her iki öğrenme eğrisinin de artık maksimum ve minimum seviyelerine yakınsadığı görülmektedir. Böylece öğrenme ve doğruluma işlemleri için ve Çizelge 4.3 ile Çizelge 4.4 sonuçları neticesinde de iterasyon miktarının uygun olduğu ve aşırı öğrenme eğilimi göstermediğine ulaşılabilir [63].



Şekil 4.10. Test verisi için çizelge 4.3'te bulunan 10. işlemin karmaşıklık matrisi.

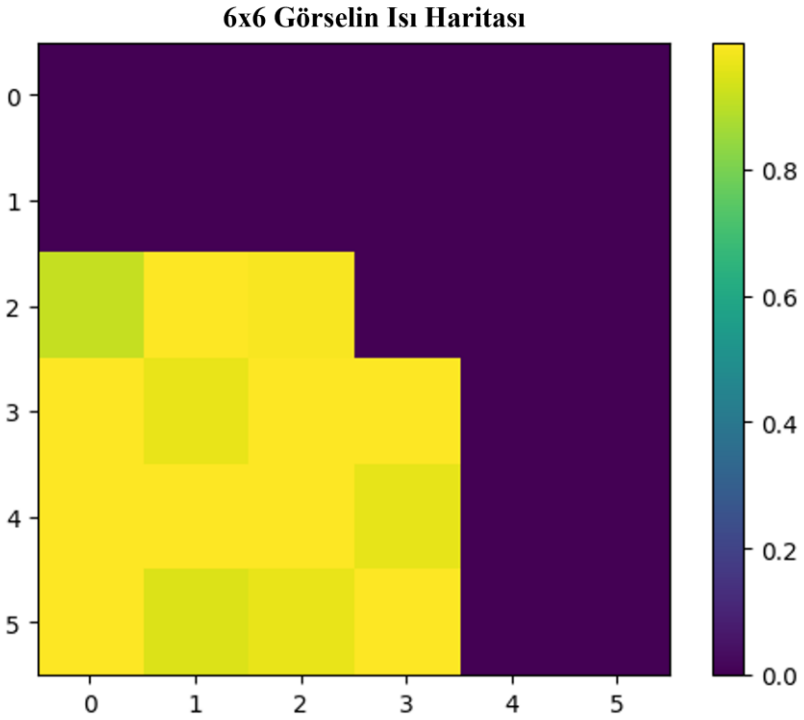
Şekil 4.10'da bulunan karmaşıklık matrisi Çizelge 4.4'deki 10. işleme ait olup, test verisinde bulunan her bir sınıfın tahmin edilen ve gerçekte olan sonuçlarının gözlemlenmesini sağlar. Şekil 4.10'da görüldüğü üzere çoğunlukla (%90 üzerinde) tahmin edilen sınıf aynı zamanda gerçekte olan sınıftır. Karmaşıklık matrisi incelendiğinde, toplamda 1200 ve her bir sınıfta 300 veri seti toplandığı için AI modeli hem gürültüyü hem de hatalı toplanan veri setlerini doğru tahmin edebildiği gözlemlenmektedir. Şekil 4.10'da bulunan karmaşıklık matrisi incelendiğinde genelde yanlış tahminlerin gerçekte olan sınıftan sonraki sınıfa ait oluştuğu gözlemlenmiştir. Bu durumun detaylı analizi için test veri seti üzerinden yanlış tahmin sonuçları incelenmiştir. Yanlış tahmin sonuçları incelendiğinde genelde benzer yapılardaki verilerde bu durum gözlemlenmiştir. Yanlış tahmin sonuçlarındaki bazı yapılar, gerçek sınıfın veri setine benzemesine rağmen gürültüden kaynaklı AI modelin yanlış tahmin sonucu ürettiği düşünülmektedir. Yanlış tahmine neden olan verilerin sayıları da toplam veri setine kıyasla da az olduğu için AI modeli yanlış tahmin etmektedir. Bunun için bu veri setlerinin de sayısının artırılması veya veri setinden çıkarılması gerektiğine ulaşılmıştır.



Şekil 4.11. Test veri için çizelge 4.6'da bulunan 10. işlemin örnek doğru tahmin sonuçları.

Hem test hem de eğitim sonucunu daha detaylı incelemek için, her iterasyonda her bir sınıftan kaç tane veri setinin iterasyona dahil edildiği algoritma yazılmıştır. Çünkü eğitim veri setinin uygun oranlarda karıştırılması, eğitimin doğruluk performansını artırırken, AI modelinin ezberleme ihtimalini azaltır [67]. Algoritma çıktı olarak ortalama her bir sınıfın %23,33-%26,67 değerleri arasında iterasyonlara dahil edildiğini fakat bazen bu oranın %40-%10 seviyelerinde olduğu bilgisini sunmuştur. Bu durumun iterasyon geçişlerinde doğruluk değerinin az da olsa inişli-çıkışlı olmasına neden olduğu düşünülmektedir [68]. Bu tarz değişimlerin sürekli oluşmasını engellemek için doğrulamanın kayıp değerine bakılarak, eğitim sırasında öğrenme oranını düşürecek fonksiyon AI modeline eklenmiştir. Eğer öğrenme eğrisinde salınım olur ise öğrenme oranı dinamik biçimde güncellenecektir [68]. Böylece eğrilerde görülebilecek salınımların önüne geçilmiştir.

Çizelge 4.6'da bulunan 10. işlem incelendiğinde herhangi bir öğrenme oranı güncellenmesi ile karşılaşılmamıştır. Test sonrası detaylı analiz için rastgele seçilen sınıfların doğru tahmin oranı ve tahmin edilen sınıf Şekil 4.11'de gözlemlenebilir. Şekil 4.11 gözlemlendiğinde farklı sınıflara ait veriler uygun ölçülerde de olmamasına rağmen AI modeli tahmin edebilmiştir. Veri setinin artırılarak, AI modelinin genelleme kabiliyetinin de arttığı test veri seti kullanılarak gözlemlenmiştir.



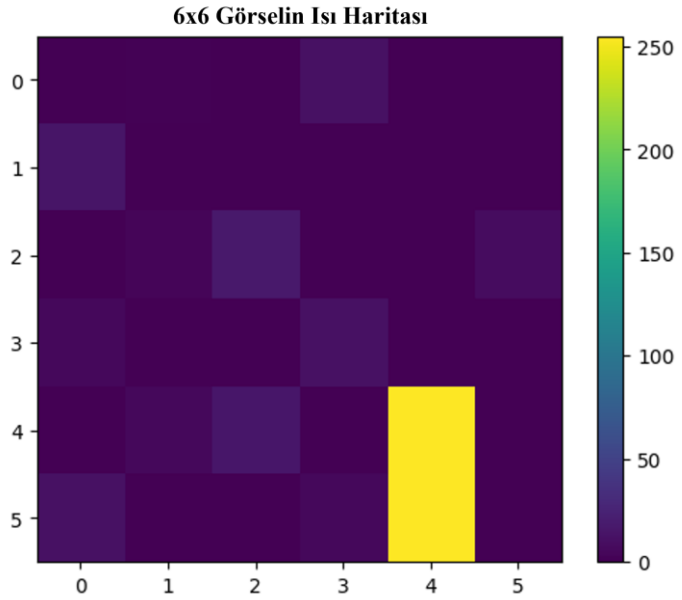
Şekil 4.12. İki doğrulama modelini karşılaştırmak için sınıf 3'e ait test görseli.

Sayısal tasarımın sentez öncesi doğrulama sonuçları ve yüksek seviyede dil ile tasarlanan doğrulama modeli de karşılaştırılarak sayısal tasarımın hassasiyeti incelenmiştir. Şekil 4.12'deki görsel sınıf 3 etiketine sahiptir. Sınıf 3 etiketi, aynı anda 16 tane CPS'ye, el hareketi yaklaştırılarak toplanan görsellerden oluşur. Şekil 4.12'deki görsel, Python ile tasarlanan doğrulama modeline ve sayısal tasarıma ait doğrulama modeline girdi olarak sürülmüştür. İki doğrulama modeli için sınıfın tahmin oranlarına Çizelge 4.5 incelenerek ulaşılabilir.

Çizelge 4.5. Doğrulama modellerinde sınıf 3 için en yüksek tahmin sonuçları.

Doğrulama Modeli	16 Bit Sabit Nokta Değeri	Ondalık Değeri
Python Modeli	0x2C51	0,3462
Sayısal Tasarım Modeli	0x300C	0,3753

Çizelge 4.5'te iki tahmin arasında 0,0291 gibi bir değer farkı görülmektedir. Tahminler arasındaki farklılığın, softmax fonksiyonunun sayısal tasarıma aktarılmasından ve sayısal tasarımda kullanılan 16 bitlik sabit nokta sayı formatının seçilmesinden kaynaklandığı düşünülmektedir.



Şekil 4.13. İki doğrulama modelini karşılaştırmak için sınıf 0'a ait test görseli.

Diğer karşılaştırmalı örnek ise Şekil 4.13'te bulunan sınıf 0 etiketine sahip test verisidir. Test görselinin sonuçları Çizelge 4.6'da verilmiştir.

Çizelge 4.6. Doğrulama modellerine sınıf 0 için en yüksek tahmin sonuçları.

Doğrulama Modeli	16 Bit Sabit Nokta Değeri	Ondalık Değeri
Python Modeli	0x25B6	0,2946
Sayısal Tasarım Modeli	0x24F4	0,2887

Çizelge 4.6 incelendiğinde bu karşılaştırma sonucunun 0.0059 gibi bir farka kadar düştüğü görülmektedir. Çünkü sayısal tasarımda bulunan softmax fonksiyonu, yüksek seviyeli dil ile tasarlanan softmax fonksiyonu kadar anlamlı rakam içermediği için her tahmin sonucu için oran farklı olmaktadır. Ek olarak, sınıf tahmin sonucunu olası etkileyecek sayısal tasarımlardan biri de bit aşımı koruması tasarımıdır [69]. Sayısal tasarımda 16 bit sabit nokta ile çalışma yapıldığı için sayısal tasarım için aritmetik işlemler sonucunda oluşabilecek bit aşımı durumuna dikkat edilmelidir [69].

Örnek olarak 0xDCEFEF62 (-0.5478) ve 0x9DD19E00 (-1.534) sayıları ondalık tabanda negatif sayıya karşılık gelmektedirler. Bu sayılardan ilki, sıradaki girdi ve ağırlık değerinin çarpımından gelmiş olup, diğer değer ise iteratif toplama sonucu oluşan değerdir. Bu nedenle ilgili değerler 32 bit boyutunda saklanmaktadır. Eğer bit aşımı koruması olmasaydı bu iki sayının toplamı 0x7AC18D62 (1.918) olarak gelecektir. İki negatif sayının toplamının pozitif sayı gelmesi bit aşımı probleminden kaynaklanır. Bu sebepten dolayı bit aşımı koruma tasarımı toplama sonucu 0x8000000 gelecektir.

Çizelge 4.7. FPGA'da gerçekleştirme sonrası kaynak kullanım sonuçları.

Kaynak Adı	**Kullanılma Miktarı**	**Kullanılma Yüzdesi**
LUT	5392	%10
FF	3184	%3
BRAM	1	%1
DSP	88	%40

Çizelge 4.7'de bulunan değerlerin elde edildiği tümleşik devre, *xc7z020clg400-1* isimli AMD'nin Zynq ailesine ait FPGA SoC (İng. system on chip) donanımıdır [70]. *Xc7z020clg400-1* modeli 28nm teknolojiyi kullanır ve toplamda 53200 LUT, 106400 FF, 140 BRAM (İng. Block RAM) ve 220 tane DSP sayısal devre elemanı içerir [70]. Zynq-7000 SoC, literatürdeki çalışmalarda da yapay zeka yongası tasarımı için kullanılan bir sistemdir [71]. Hem FPGA hem de ARM mimarili bir çekirdek içermesinden dolayı herhangi bir ek donanıma ihtiyaç duyulmadan birçok işlem yükünü gerçekleştirme kabiliyetine sahiptir [70, 71].

Çizelge 4.7 incelendiğinde düşük miktarda kaynak kullanılarak sayısal tasarımın gerçekleştirildiği gözlemlenmektedir. Diğer kaynaklara kıyasla DSP kullanımının fazla olmasının temel nedeni her bir düğümde yapılan çarpma işlemi ve çarpma işlemi sonuçlarının toplamından kaynaklıdır. Bundan dolayı her bir düğüm 2 tane DSP kullanmaktadır [72]. Çizelge 4.7'de bulunan BRAM, softmax fonksiyonunun değerlerini saklamak için kullanılmaktadır.

Çizelge 4.8. FPGA'da gerçekleştiren MLP mimarisinin güç tüketimi sonuçları.

Güç Tüketim Türü	**Güç Tüketim Değeri (mW)**
Statik	108
Dinamik	21
Toplam	129

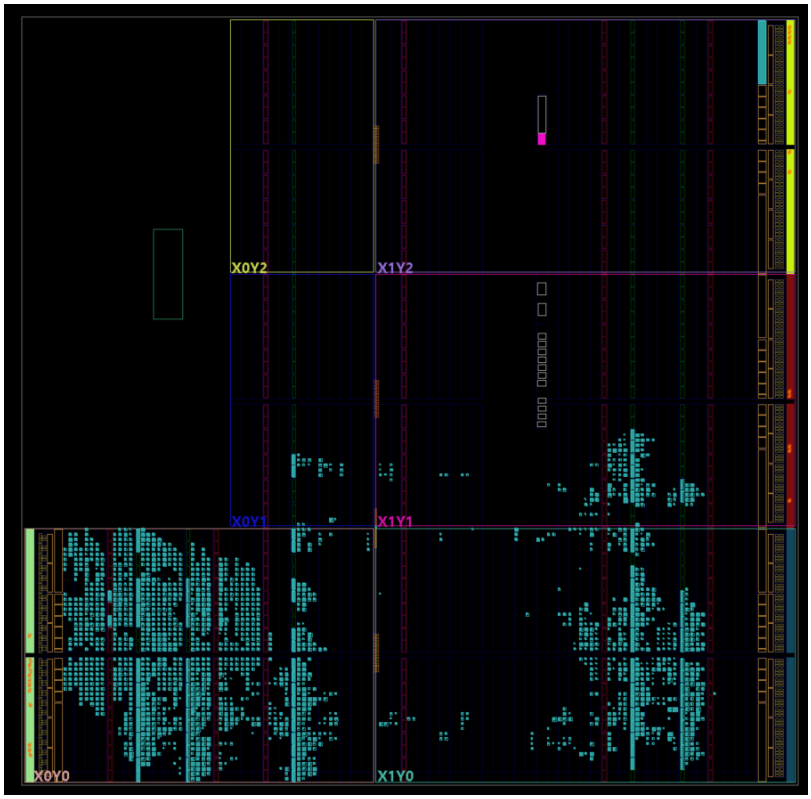
Çizelge 4.8'de FPGA'da gerçekleştirilen MLP mimarisinin güç tüketim değerleri görülmektedir (Şekil Ek.2). Vivado 2023.2.2 yazılımı ile güç analizi yapılırken, MLP mimarisi ile sayısal tasarımda kullanılan PLL'nin (İng. Phase Locked Loop) güç tüketim değerleri ayrılmıştır. Çünkü PLL donanımı saat frekansı üretmek ve üretilen saat frekansının kararlılığı sağlamak için kullanılan bir donanımdır. FPGA'den zaman ve güç raporu oluşturmak için PLL kullanılmıştır. Fakat PLL kullanılması son tasarımda zorunlu değildir. Bundan dolayı güç tüketiminde hesaplamaya eklenmemiştir.

Çizelge 4.9. FPGA'da gerçekleştirilen MLP mimarisinin zaman kısıt sonuçları.

Zaman Parametresi	**Zaman Değeri**
Kullanılan Saat Frekansı ve Periyodu	12,5 MHz ve 80 ns

Çizelge 4.9 incelendiğinde FPGA'da gerçekleştirilen MLP mimarisinin zaman kısıt sonuçlarına ulaşılabilir. Kullanılan saat frekansı zaman kısıt raporlarının incelenmesi sonucunda, kritik yolun (İng. critical path), softmax fonksiyonunun içerdiği normalize denkleminde görülmüştür. Farklı AI uygulamalarında, softmax fonksiyonu yerine işlem maliyeti daha düşük aktivasyon fonksiyonu tercih edilerek daha yüksek hızlara ulaşılabilir [57].

Çizelge 4.8 ve Çizelge 4.9'da sunulan FPGA'ya ait parametre değerleri ile 5. bölümde Yapay Zeka Yongası Tasarımı başlığı altında ASIC'ye ait parametre değerleri ile kıyaslanacaktır. Sayısal tasarımın gerçekleştirilmesi sonucu oluşan kaynak kullanımının dağılımına Şekil 4.14'den ulaşılabilir.

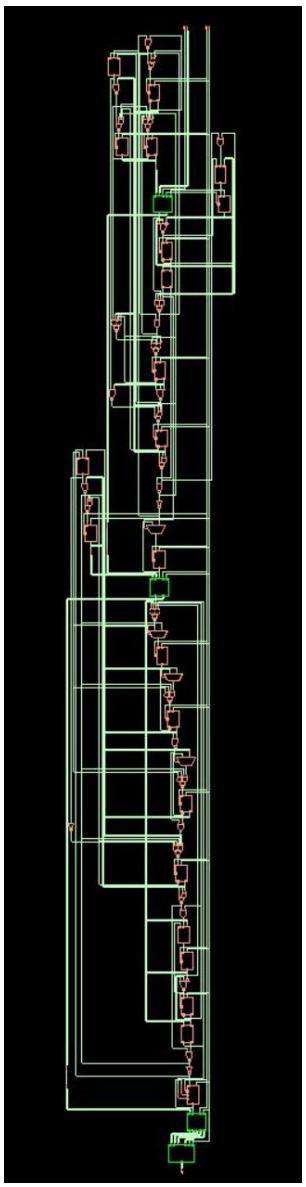


Şekil 4.14. Sayısal tasarımın FPGA'da gerçekleştirilmesi sonucu kaynak kullanım dağılımı.

5. YAPAY ZEKA YONGASI TASARIMI

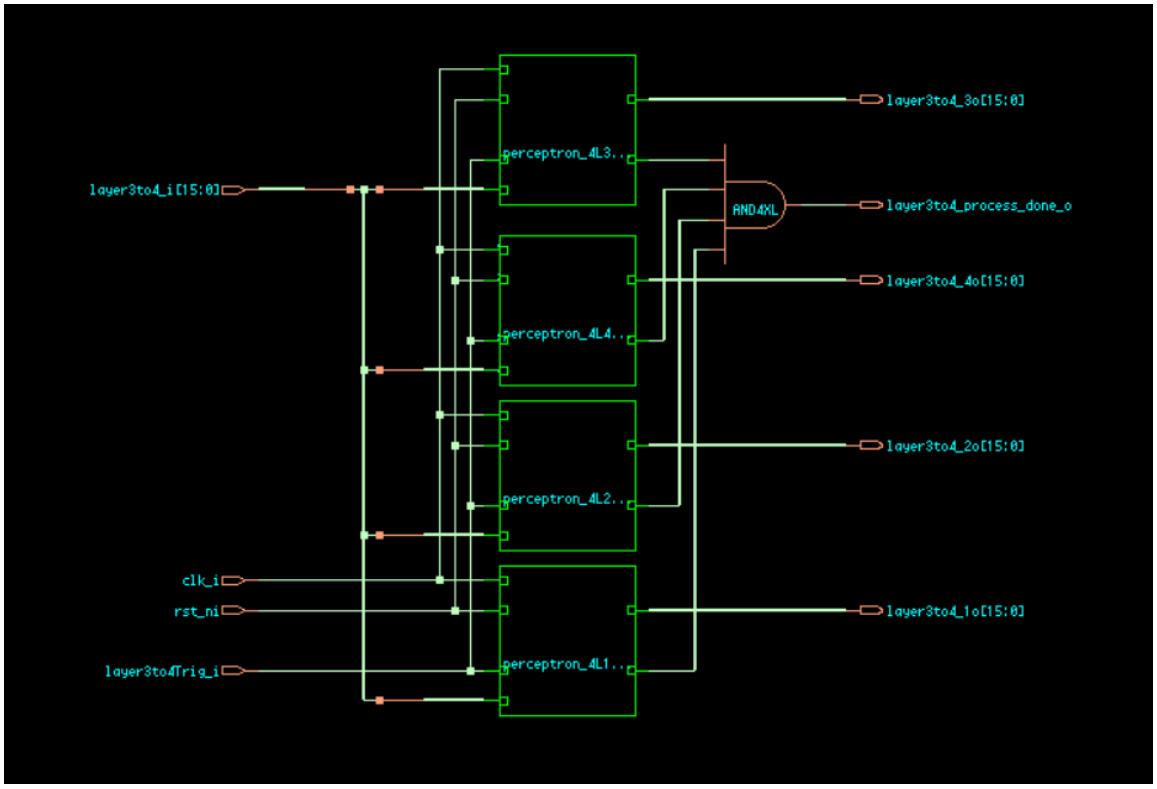
5.1. MLP Mimarisinin ASIC Tasarım Aşamaları

MLP modeli, FPGA tümleşik devresinde doğrulanıp, gerçekleştirildikten sonra yapay zeka yongası tasarımına başlanılmıştır. Yapay zeka yongası tasarımında, Şekil 3.4 görselindeki akış diyagramı takip edilmiştir. Yapay zeka yongası tasarımında sentez aşamasında Cadence Genus ve gerçekleştirme aşamasında ise Cadence Innovus yazılımı kullanılmıştır [73]. Genus, RTL sentezi ve fiziksel sentez çözümü sağlayan bir yazılımdır [74].



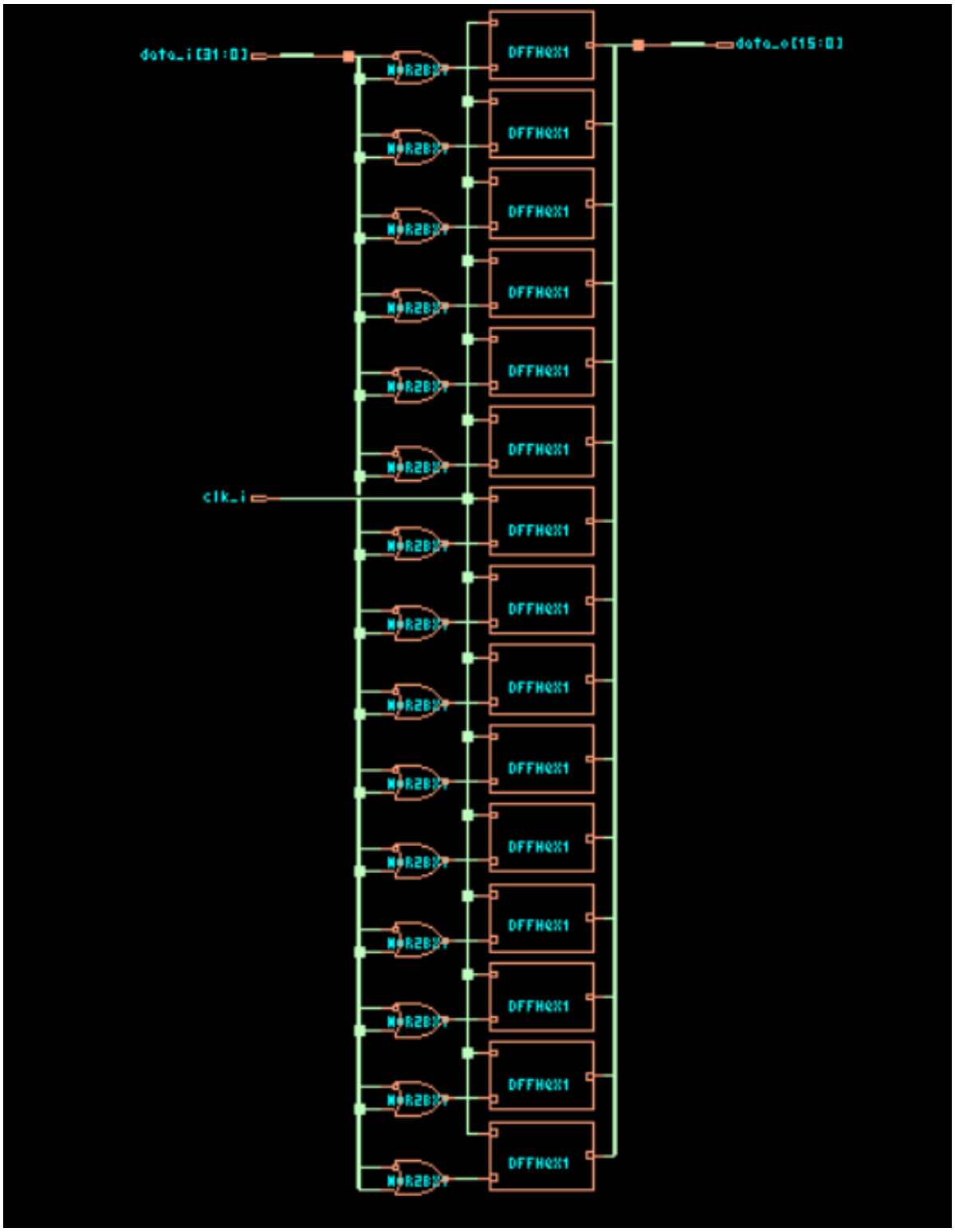
Şekil 5.1. Yapay zeka yongasının blok hücre diyagramı.

Sentez yapıldığı sırada zaman gereksinimi, yonganın alanı ve güç gereksinimleri arasında öncelik belirlenebilmektedir [73]. Yapay zeka yongası için bütün öncelikler eşit seviyede belirlenmiştir. İsterlere bağlı olarak ASIC tasarımının güç ve kaynak tüketimi artırılarak daha yüksek frekans değerlerine ulaşılabilir. Şekil 5.1'de MLP mimarisinin blok hücre diyagramına ulaşılabilir.



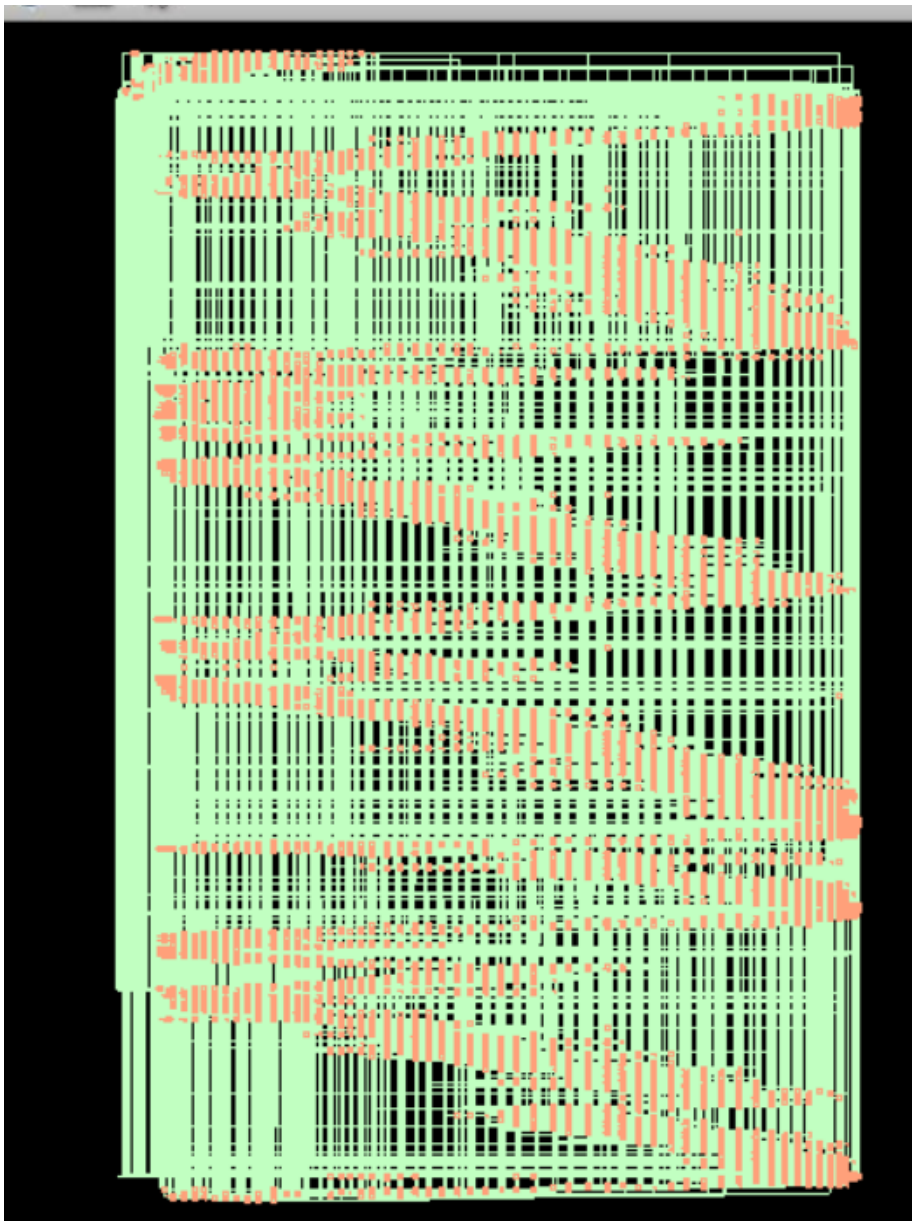
Şekil 5.2. Çıktı katmana ait blok hücre diyagramı.

Şekil 5.2'deki tasarım, Şekil 3.13'teki 4. Katman yani çıktı katmanına ait tasarımdır. Şekil 5.3'deki blok hücreler incelendiğinde, Şekil 3.13'teki gibi 4. katmana saat, sıfırlama, tetik işareti sürüldüğü görülmektedir. Ek olarak 3. katmanın çıktılarının paralel biçimde sürüleceği girdi veri yolu da Şekil 5.2'de görülebilir. ReLU ve Softmax aktivasyon fonksiyonları düğümlerin içerisinde bulunmaktadır. Softmax fonksiyonu, sadece 4. katmandaki düğümlerde bulunmaktadır. ReLU ise 4. katmanın dışında kalan bütün düğümlerde bulunmaktadır. Aktivasyon fonksiyonlarının blok diyagramları tamamen kapı seviyesi hücrelerden oluşmuştur. Aktivasyon fonksiyonlarının tamamen kapı hücresinden oluşan ReLU Şekil 5.4'te, Softmax ise Şekil 5.5'te sunulmuştur.



Şekil 5.3. ReLU fonksiyonun blok hücre diyagramı.

Şekil 5.3 incelendiğinde sentezleyici tarafından ReLU fonksiyonu sadece NOR kapısı ve DFF hücresi kullanılarak oluşturulmuştur.



Şekil 5.4. Softmax'in normalizasyon işlemine ait blok hücre diyagramı.

Şekil 5.4 incelendiğinde Softmax fonksiyonu, ReLU fonksiyonuna göre hem çok daha fazla hücre ile oluşturulmuştur hem de daha karmaşık bir yapıdadır. Çünkü Softmax tasarımı, girdi değerine bağlı olarak çıktı değeri sabit olan ve literatürde de LUT tabanlı tasarım olarak [57] bilinen tasarım yöntemi tercih edilmiştir. Softmax fonksiyonunda sayısal tasarımında öncellikle girdinin üstel değeri, LUT'a gömülen değerler üzerinden üretilir. Sonrasında Şekil 5.4'te blok hücre diyagramı bulunan normalizasyon yapılır. En son olarak bütün değerler karşılaştırılır ve en yüksek değer tahmin sonucu olarak dışarıya sürülür. Softmax fonksiyonu çıktı olarak 16 bit sabit nokta formatında sayı üretir. Bu

tasarım tercihlerinden dolayı üstel fonksiyonun değerleri 2kB'lık bilgi saklamaktadır. Softmax fonksiyonunun FPGA'da 1 BRAM kullanmasının nedeni de bu durumdan kaynaklıdır. Kullanılan sentezleyici sürümünün hazır kütüphanelerinde herhangi bir SRAM veya ROM hücresi bulunmadığı için sentezleyici, softmax fonksiyonunu tamamen kapı seviyesi oluşturmuştur. Gerçekleştirme aşamasında kullanılacağı için, sentezleme aşamasında oluşturulan devre bağlantı listesi ve tasarım kısıtlama dosyaları kaydedilmiştir. Bu aşamanın ardından yapay zeka yongasının gerçekleştirilme aşamasına geçilmiştir.

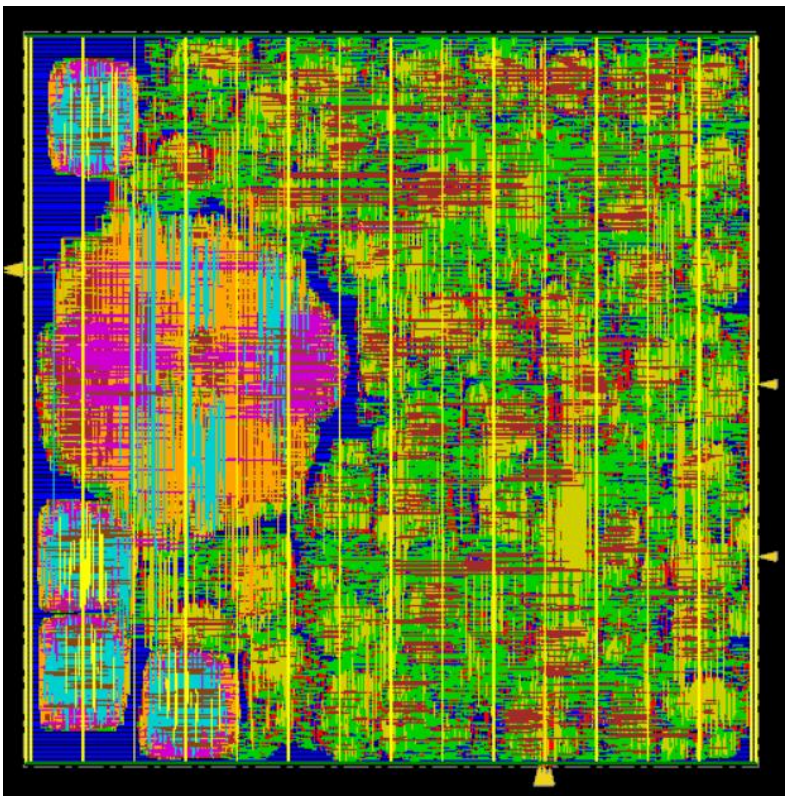
5.2. Gerçekleştirme Aşaması

Yapay zeka yongasını gerçekleştirmek için Cadence Innovus yazılımı kullanılmıştır. Innovus, sentezleme sonrası devre bağlantı listesi ve zaman kısıtı gibi dosyaları kullanarak sayısal tasarımın fiziksel olarak yerleşimini ve bağlantılarını sağlayan bir fiziksel yonga gerçekleştirme ortamıdır [76]. Daha önce detaylandırılan ASIC tasarım akışına uyularak, yapay zeka yongası gerçekleştirilmiştir. Fiziksel yonga tasarımı için Cadence'ın 45nm standart hücre kütüphanesi kullanılmıştır. Kütüphaneler çağrıldıktan sonra katman planlaması aşamasına geçilmiştir.

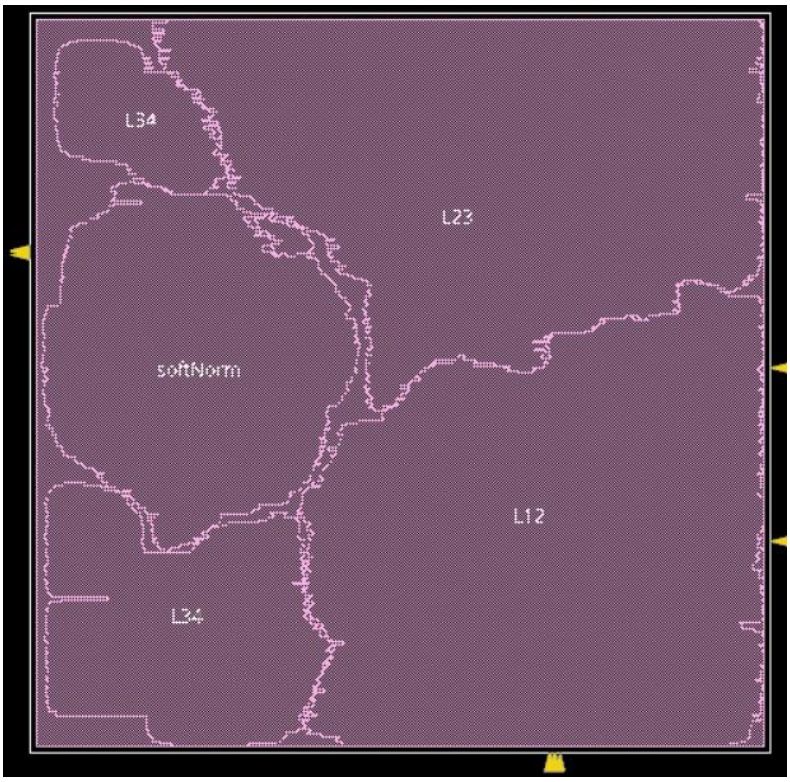
Çizelge 5.1. Serimde katman planlaması için kullanılan parametreler.

Parametre	**Değer**
Yükselik/Genişlik Boyutları	500 / 500 μm
Çekirdek ile I/O arası pay farkı (Her yerden aynı seviyede)	4 μm
Yüzük için Yukarı ve Aşağı Katmanda Kullanılan Metal ve Ölçüleri	Metal-11; 0,5 / 0,2 μm
Yüzük için Sol ve Sağ Katmanda Kullanılan Metal ve Ölçüleri (W/S)	Metal-10; 0,5 / 0,2 μm
Şerit'in Kullanıldığı Katman; Ölçüleri (W/S)	Metal-10; 0,3 / 0,3 μm
Toplam Kullanılan Şerit Miktarı	15

Çizelge 5.1'de katman planlamasında ayarlanan parametreler görülmektedir. Fiziksel bileşenlerin parametreleri ayarlandıktan sonra veri bağlantı dizisi fiziksel olarak, planlanan katmana aktarılmıştır. Eklenen fiziksel parçaların birbirleri ile bağlantıları tamamlandıktan sonra CTS (İng. Clock Tree Synthesis) öncesi iyileştirme aşamasına geçilmiştir. CTS öncesi iyileştirme aşaması sonucunda elde edilen katman planı ve amip görüntüsü sırasıyla Şekil 5.5 ve Şekil 5.6'da sunulmuştur.

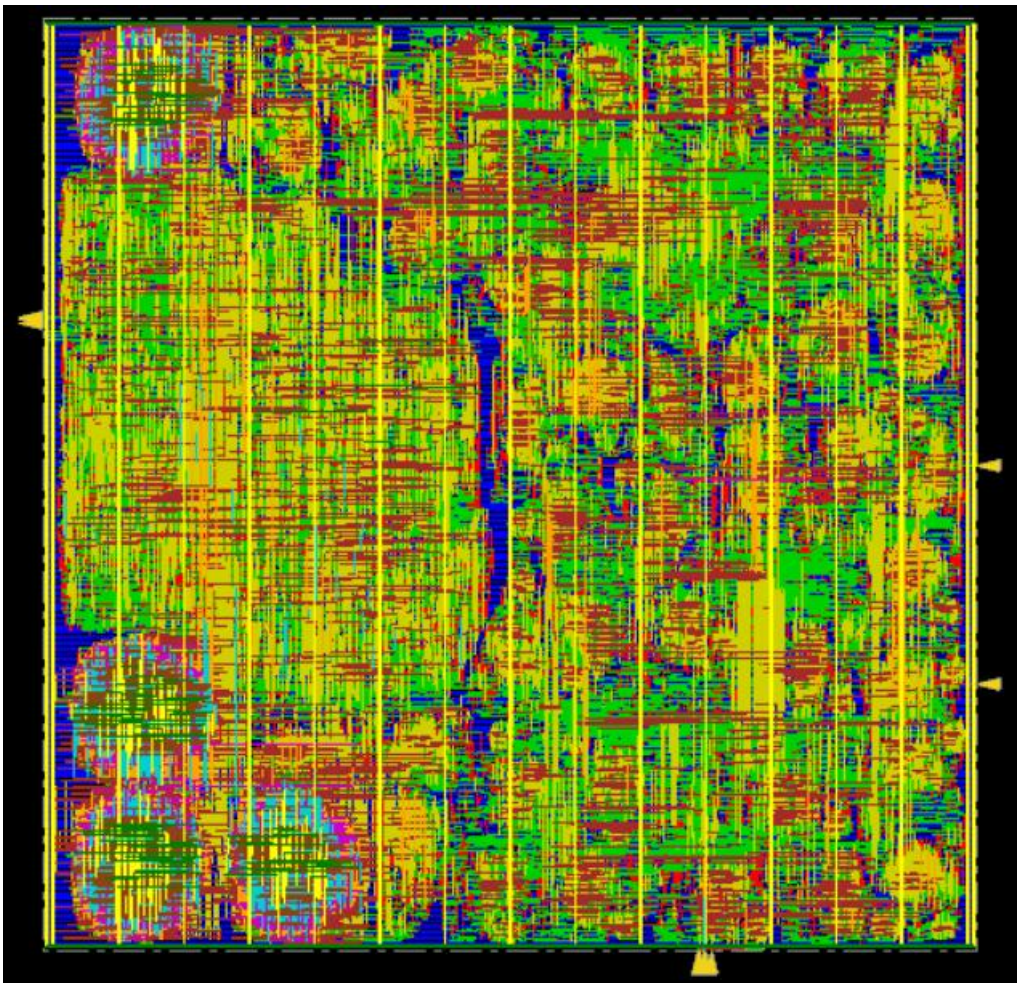


Şekil 5.5. İyileştirme akışı öncesi elde edilen katman planı görünümü.



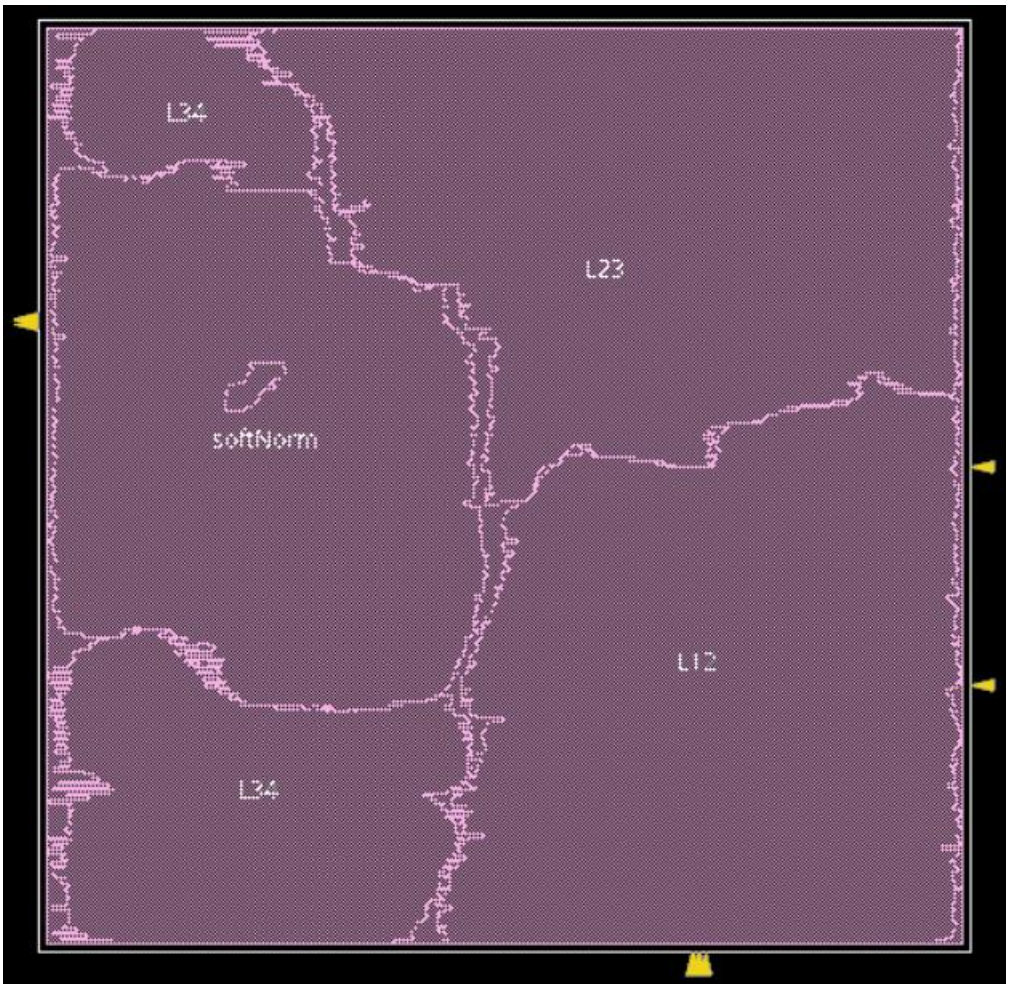
Şekil 5.6. İyileştirme akışı öncesi elde edilen yonga amip görünümü.

Şekil 5.5 ve Şekil 5.6'da yapay zeka yongasına ait farklı serim görünümlerine ulaşılabilir. Softmax fonksiyonunda bulunan normalizasyon işlem bloğu Şekil 5.6'da *softNorm* ismiyle görülmektedir. Kapı seviyesinde bölme işlemi yapmak çok maliyetli olduğu için Şekil 5.4'ten anlaşılacağı üzere devre bağlantı listesi (İng. netlist) büyük boyutludur. Şekil 5.5 incelendiğinde dört tane kareye yakınsayan şekilde blok görülmektedir. Bu bloklar üstel değerlerin gömüldüğü hücre bloklarıdır. MLP mimarisinin çıktı katmanında dört tane düğüm olduğu toplamda dört tane üstel değerlerin gömüldüğü hücre bloğu bulunur. Halen herhangi iyileştirme aşaması uygulanmadığı için, katman planından çok fazla boşluk bulunmaktadır. İyileştirmelerden sonra serim, kısıtlar dahilinde en uygun hale getirilerek bütün ihlaller giderilir.

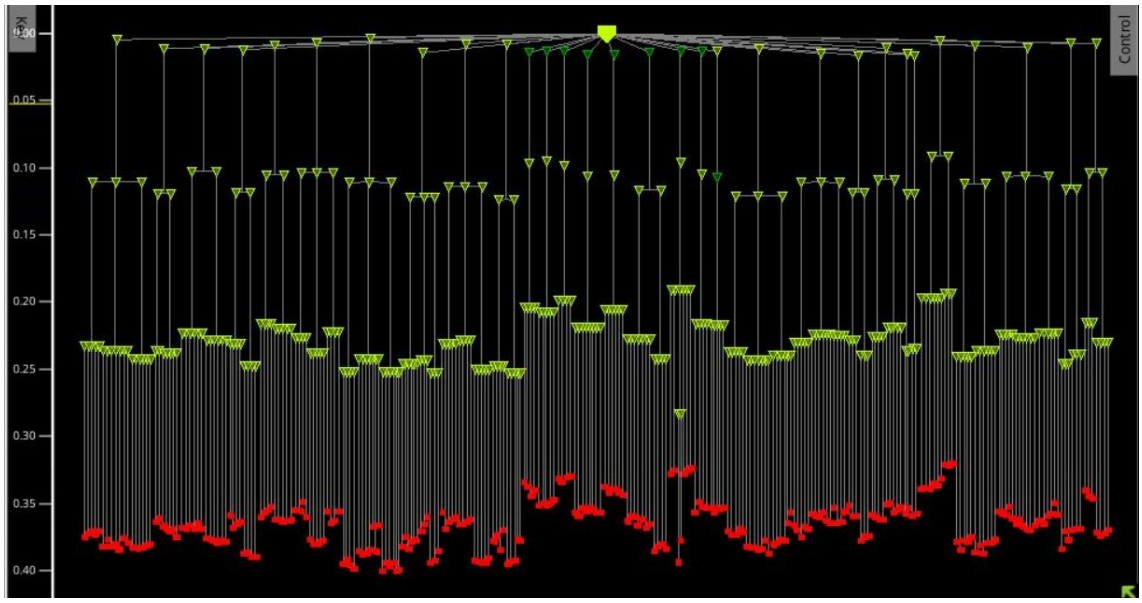


Şekil 5.7. CTS öncesi iyileştirme sonucunda elde edilen katman planı görünümü.

Şekil 5.7'de görüldüğü üzere serim artık katman planı üzerine daha da dağılmıştır. Böylece yapay zeka yongasının katman planının kullanım yoğunluğu artırılarak bu aşama için ihlaller giderilmiştir. Şekil 5.8'deki serimin katman planı üzerinde dağılımı, Şekil 5.6 ile karşılaştırılarak incelenebilir.

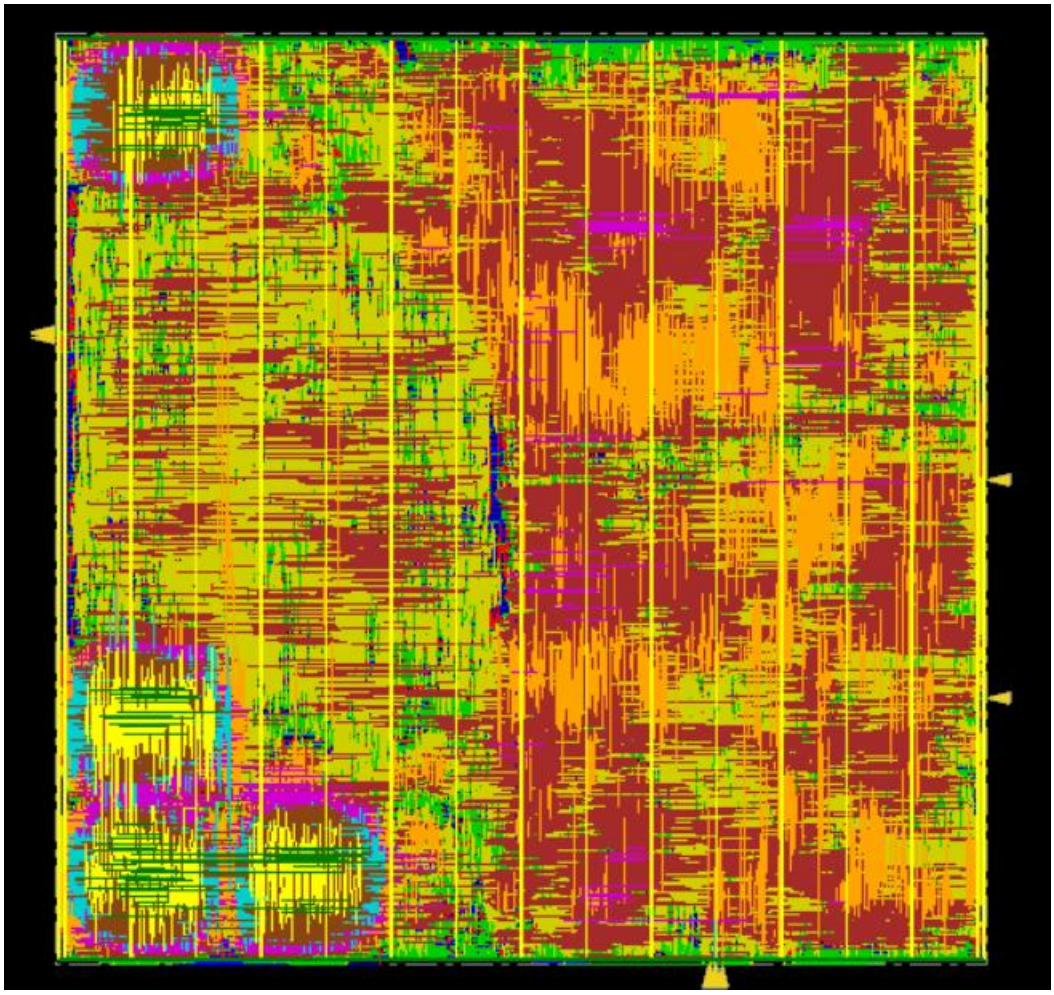


Şekil 5.8. CTS öncesi iyileştirme sonucunda elde edilen amip görünüm.



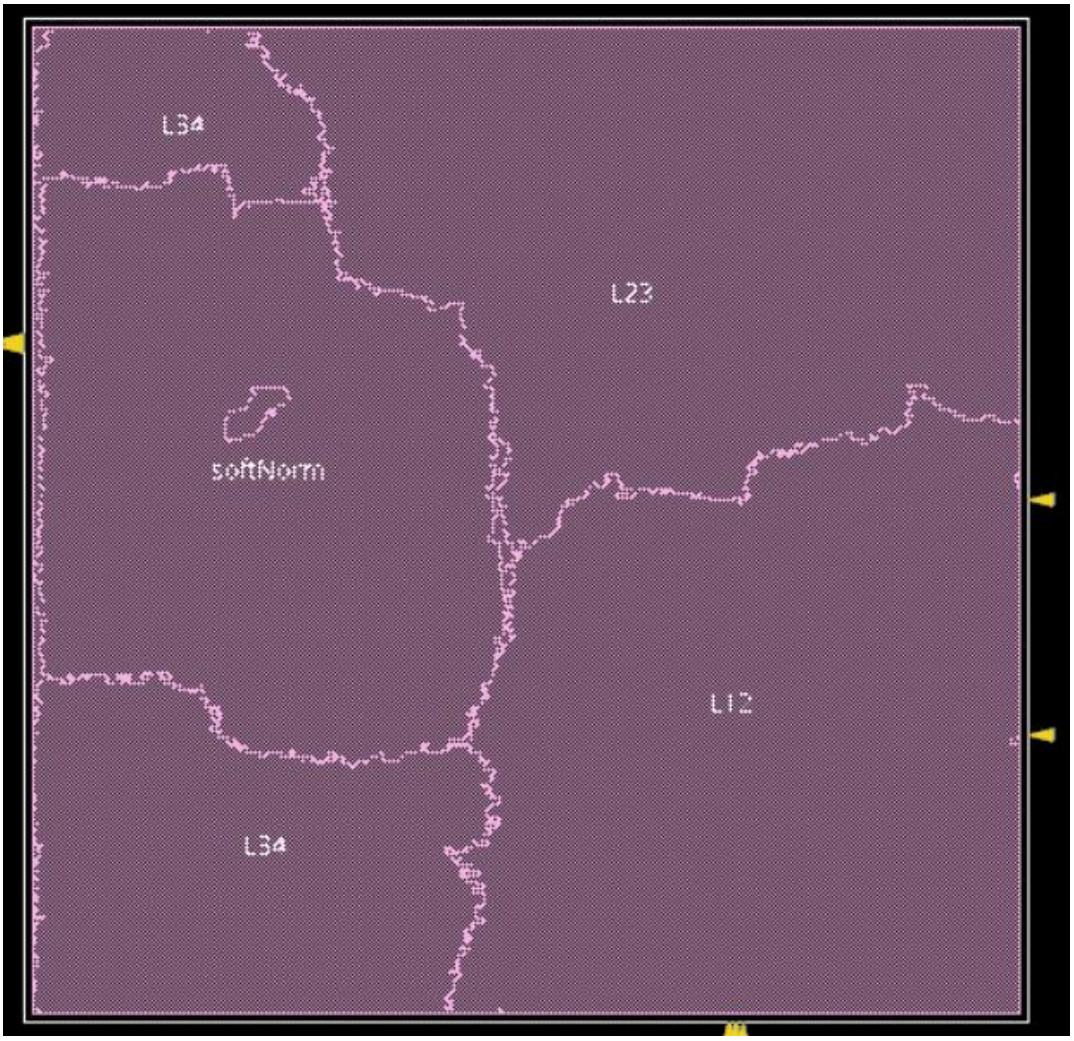
Şekil 5.9. Serim üzerinde saat ağacının dağılımı.

Şekil 5.9'de oluşturulan saat ağacının dağılımı görülebilir. CTS oluşturulduktan sonra zaman kısıtının raporunun çıkarılması gerekir. Zaman raporunda ihlaller bulunursa CTS sonrası iyileştirme aşamasına geçilir. Bu aşamada herhangi bir zaman ihlali, zaman kısıt raporu ile denetlenebilir. İhlaller giderilene kadar, sıradaki aşamaya geçilmemiştir ve bağlı iyileştirmelere devam edilmiştir. CTS sonrası ve bağlantı sonrası iyileştirmeler, zaman ihlalleri giderilene kadar koşturulmaya devam edilir. Zaman ihlalleri giderildikten sonra tasarım kural denetimi (İng. Design Rule Check – DRC), bağlantısallık ve anten ihlali oluşmamasını doğrulayacak denetimler koşturulur. Denetim aşamalarında da herhangi bir ihlal ile karşılaşılmadığı için ince ayar bağlantı iyileştirmesi koşturulmasına gerek kalmamıştır. Dolgu maddesi (İng. filler) de fiziksel olarak yerleştirildikten sonra yapay zeka yongasının gerçekleştirilme aşaması sonlanmıştır. Şekil 5.10 ve Şekil 5.11'te sırasıyla yapay zeka yongasının nihai katman planlaması ve amip görünümüne ulaşılabilir.



Şekil 5.10. Yapay zeka yongasının nihai katman planı görünümü.

Yapay zeka yongası, Çizelge 5.1'de bulunan parametre değerleriyle katman planının %90,335 yoğunlukta doldurmaktadır. Güç, zaman ve alan arasında uygun ister öncelikli gerçekleştirme yapıldığında yoğunluk değeri değişecektir. Böylelikle tez çalışmasının amacı olan yapay zeka yongası tasarımı gerçeklenmiştir.



Şekil 5.11. Yapay zeka yongasının nihai amip görünümü.

Yapay zeka yongasının gerçekleştirilmesi bitirildikten sonra Innovus yazılımı üzerinden serimin RTL2GDSII dosyası oluşturulmuştur.

5.2. MLP Mimarisinin Farklı Tümleşik Devrelerde Gerçekleştirilme Sonuçlarının Karşılaştırılması

MLP Mimarisinin ASIC olarak gerçekleştirildikten sonra elde edilen sonuçlar Çizelge 5.2 ve Çizelge 5.3'te sunulmuştur. Bu sonuçlar MLP mimarisi FPGA SoC kullanılarak gerçekleştirdiğindeki sonuçlar ile karşılaştırılacaktır.

Çizelge 5.2. MLP mimarisinin ASIC tasarımının güç tüketimi sonuçları.

Modül İsmi	**Toplam Güç Tüketimleri (mW)**
Statik (Sızıntı)	0,005
Dinamik (Anahtarlama)	2,901
Toplam	2,906

Çizelge 5.2 incelendiğinde statik ve dinamik olmak üzere ayrı güç tüketim değerlerine ulaşılabilir. Tablo incelendiğinde sızıntı ve dinamik güç tüketimi arasındaki fark çok fazladır. Çünkü tamamen özel ASIC tasarımı ile sadece uygulamaya özgü problemi çözmek için gerekli mantıksal elemanlar kullanılabilir. MLP mimarisini hem FPGA hem de ASIC'de karşılaştırmak için dinamik güç tüketimini incelemek gerekir. Çizelge 4.8 ve Çizelge 5.2 karşılaştırıldığında MLP mimarisi FPGA SoC'de gerçekleştirildiğinde ASIC'ye kıyasla yaklaşık 7,25 kat daha fazla dinamik güç tüketeceği öngörülmektedir.

Çizelge 5.3. MLP mimarisinin ASIC tasarımının zaman kısıt sonuçları.

Zaman Parametresi	**Zaman Değeri**
Kullanılan Saat Frekansı ve Periyodu	50 MHz ve 20 ns

Çizelge 5.3 incelendiğinde ASIC'de gerçekleştirilen MLP mimarisinin zaman kısıt sonuçlarına ulaşılabilir. Çizelge 4.9 ve Çizelge 5.3 karşılaştırıldığında MLP mimarisi FPGA SoC'de gerçekleştirildiğinde ASIC'ye kıyasla 4 kat daha düşük saat frekansına

sahip olacağı öngörülmektedir. Bunun sonucunda ASIC tasarımının, FPGA SoC tasarımına kıyasla 4 kat daha hızlı sonuç üretebilir. ASIC'de sayısal tasarım gerçekleştirmenin, FPGA SoC'de gerçekleştirmeye kıyasla bu kadar düşük güç tüketimi yapay zeka tümleşik devresi tercihinde ASIC kullanılmasının nedenlerinden biri olmaktadır [75]. MLP modelinin, sayısal tasarımına ilk girdi tetiği sürüldükten 101 saat darbesi sonra tahmin sonucunu üretmektedir. Tümleşik devrelerde kullanılacak saat frekansları ile hesaplama yapıldığında MLP modelinin tahmin sonucu, FPGA'da 8,08µs ve ASIC'de ise 2,02 µs'de üretilecektir.

6. SONUÇLAR VE YORUMLAR

Tez çalışmasında, CPS dizilerinden oluşturulan veri seti kullanan AI modeli ile ağırlık ve sapma değerleri elde edilmiştir. AI modelinde kullanılan aynı MLP mimarisi de sayısal tasarıma aktarılarak yapay zeka yongası tasarlanmıştır. Tasarım sürecine ilk olarak, tez çalışmasıyla uyumlu olarak hazır veri setleri kullanılarak AI modelinin doğrulanmasına ve olgunlaşmasına başlanmıştır. Bu çalışmada ek olarak MLP mimarisini de içeren AI modelinin hiper parametreleri güncellenerek sonuçlarının kapsamlı ve karşılaştırılmalı analizi gerçekleştirilmiştir. Sonrasında kullanılmak üzere AI modelinin en uygun öğrenim sonucunda elde edilen ağırlık ve sapma değerleri kaydedilmiştir. MLP mimarisi yüksek seviyeli dil ile tasarlanıp, ilgili ağırlık ve sapma değerleri sürülmüştür. Test verileri görüntü işleme ile analiz edilerek, MLP mimarisine sürülerek test edilmiştir ve sayısal tasarıma aktarılma süresince karşılaşılacak sorunlara çözüm üretilmeye çalışılmıştır. Elde edilen sonuçlar doğrultusunda CPS dizisinden oluşturulacak veri seti boyutu ve tipi belirlenmiştir. Oluşturulan veri seti, görüntü işleme yöntemleri kullanılarak analiz edilmiştir. Analiz sonuçlarına göre ön işleme sürecinden geçilerek AI modelinde kullanılmaya hazır hale getirilmiştir. AI modelinin eğitim çıktıları incelenerek eğitim, doğruluk ve test verilerinin sonuçları analiz edilmiştir. Analiz sonucunda her test verisi için doğruluk, kayıp, duyarlılık, kesinlik ve F1 puanı incelenerek en uygun sonuçlar elde edilmeye çalışılmıştır. Bu durumu sağlamak için AI modelinin ilgili hiper parametreleri iteratif biçimde güncellenmiştir. En uygun sonuçlar elde edildikten sonra ağırlık ve sapma değerleri kaydedilerek, yüksek seviyeli dil tasarımına aktarılıp kaydedilen ağırlık ve sapma değerleri de doğrulanmıştır.

Algılayıcı seviyesinde sayısal tasarıma aktarım için davranışsal tasarımlar yapılmıştır. Tek algılayıcı sayısal tasarıma aktarılıp, doğrulandıktan sonra MLP mimarisinin blok seviyesi davranışsal tasarımına başlanmıştır. MLP mimarisi sayısal tasarıma aktarılıp, doğrulandıktan sonra hem FPGA SoC hem de ASIC seviyesinde tasarıma aktarılarak parametreler bazında karşılaştırılması yapılmıştır. Sonuç olarak tez çalışmasında karşılaştırmalar sonucunda yapay zeka yongasının farklı entegre devreler kullanılarak tasarlanmasının avantajları ve dezavantajları hakkında kapsamlı bilgi sunmaktadır. Tez çalışması, yapay zeka yongasının tasarım süresinin her aşamasında olası problemleri ve çözümleri hakkında fikir vermektedir. Tez çalışmasında sunulan tasarım akışları ve kapsamlı analiz sonuçlarıyla, en baştan özel veri seti oluşturulup ve yapay zeka yongası tasarımına kadar süreç planlama sunan bir rehber olarak değerlendirilebilir.

Tez çalışması süresince planlanan tasarım akışı takip edilerek çalışmalara devam edilmiştir. Çalışmanın ilk aşamaları hazır veri seti üzerinden AI modelini olgunlaştırma ve doğrulama aşamasıdır. Bu aşamada tez çalışmasına tamamen uygun veri seti bulunamadığı için olabildiğince uygun veri seti bulunarak ve bazı görüntü işleme yöntemleri kullanılarak ilgili veri setlerinin, tez çalışmasında kullanılabilir hale getirilmesine çalışılmıştır. Örnek olarak tez çalışmasında toplam 36 piksel üzerinden işlem yapılmıştır. Fakat 36 piksellik veri seti bulunamadığı için olabildiğince tez çalışmasına uygun ve düşük çözünürlüklü veri seti bulunmaya çalışılmıştır. Araştırmalar sonucunda bulunan bazı veri setleri görüntü işleme yöntemleri ile 36 piksele düşürüldüğünde görüntünün ayırt edici özelliklerini kaybettiği gözlemlenmiştir. Bu problemden dolayı etiketlenmesi belirgin veri setleri üzerinden işlem yapılmıştır. Böylelikle daha uygun öğrenme sonuçları elde edilse de bu aşama da yapılan çalışmaların bazıları (örnek: Alt-örnekleme) tez çalışmasında planlamanın esnetilmesine ve güncellenmesine neden olmuştur. El hareketleri yaklaştırılarak oluşturulan veri setleri 4 farklı sınıfa ayrılmıştır. Oluşturulan veri seti, CPS dizilerinin birbirlerinden ve ortamın gürültüsünden etkilenmiştir. Bundan dolayı AI modelinin, test veri setine ait tahmin sonuçları analiz edilmiştir. Analiz sonucunda yanlış tahmin sonuçlarının, yanlış oluşturulan verilerden kaynaklandığına ulaşılmıştır. Ek olarak yanlış oluşturulan veriler, kendi sınıfındaki diğer verilere kıyasla çok daha az olduğuna da ulaşılmıştır. Bu durum da AI modelinin genelleme kabiliyetini olumsuz etkilemiştir. AI modelinin genelleme kabiliyetinin artırmak için yanlış toplanan verilerin düzeltilmesi veya eğitime dahil edilmemesi gerektiği sonucuna ulaşılmıştır. Sonuç olarak tez çalışmasında ilk başta planlanan tasarım akışı, tez süresince dinamik olarak güncellemeye devam edilmiştir. Nihai hali tez çalışmasına eklenerek, karşılaşılan problemlere karşı çözümlere aynı konu altında tez çalışmasına kapsamlı analizi ile birlikte eklenmesi uygun görülmüştür. Yüksek seviyeli dil tasarımını olgunlaştırdıktan sonra düşük seviyeli dil ile tasarıma geçiş süreci beklenmeyen birçok problemin öngörüsünü sağlamıştır. Böylelikle yapay zeka yongasına başlamadan karşılaşılacak problemlerin analizi sayesinde düşük seviyeli dil ile yapılan sayısal tasarımın sürecinde daha planlı ve hızlı ilerlendiği düşünülmektedir. AI modelinde bulunan hiper parametrelerin iteratif güncellemeleri, literatür araştırmaları sonucunda belirlenmiştir. Böylelikle literatür araştırmalarında iddia edilen güncellemelerin avantajları, tez çalışmasında gözlemlenmiştir. Ek olarak literatür çalışmaların incelenip tez çalışmasında uygulanarak elde edilen sonuçlar analiz ve test yöntemleri ile

değerlendirilmiştir. Sonuç olarak düşük zaman ve hesaplama maliyeti ile beraber uygun öğrenim sonuçlarına ulaşılmıştır.

Tez çalışması, günümüz problemlerden biri olan model tanıma ve tespiti için özgün akıllı cihaz tasarımı fikrini farklı çalışma disiplinleri sistematik biçimde birleştirerek, daha az donanım kullanımı ve daha özgün tasarım yaklaşımı ile literatüre kazandırmıştır. FPGA / ASIC tasarımı gelecekte daha kapsamlı bir yapıda tasarlanarak birden fazla özgün probleme tek bir donanım tasarımı ile çözüm sunmayı amaçlamaktadır. Örnek olarak önceden eğitilmiş modellerden elde edilen ağırlık ve sapma değerleri, probleme bağlı olarak dinamik biçimde entegre devreye sürülerek, problemlere daha esnek yaklaşım sunan bir entegre devre tasarımı yapılabilir. Bu sayede her özgün problem için farklı donanım kullanmak yerine aynı donanımın probleme bağlı olarak güncellenmesi fikri ortaya çıkmaktadır. Bu fikir ile daha düşük sayıda donanım kullanımı, donanım israfı, taşınabilir sistemlerin geliştirilmesi ve uyarlanabilir teknolojiler konusunda yaklaşımlar yapılabilir.

7. KAYNAKLAR

[1] J. Liu, et al., Artificial Intelligence in the 21st century, IEEE Access, 6: 34403-34421, doi: 10.1109/ACCESS.2018.2819688**, (2018).**

[2] W. Ertel, Introduction to Artificial Intelligence, Second Edition, Springer, **(2017).**

[3] W. Samek, et al., Explaining deep neural networks and beyond: A review of methods and applications, Proceedings of the IEEE, 109.3: 247-278, doi: 10.1109/JPROC.2021.3060483, (**2021).**

[4] H. Hussain, P.S. Tamizharasan, C.S. Rahul, Design possibilities and challenges of DNN models: a review on the perspective of end devices, Artificial Intelligence Review, 1-59, 5109-5167, https://doi.org/10.1007/s10462-022-10138-z, **(2022)**.

[5] A. Y. Bahar., et al., Survey on features and comparisons of programming languages (PYTHON, JAVA, AND C#), 2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems, pp. 154-163, doi: 10.1109/ICETSIS55481.2022.9888839, **(2022).**

[6] S. M. M. Ahsan., et al., Hardware Accelerators for Artificial Intelligence, In: Iranmanesh, A., Sayadi, H. (eds) AI-Enabled Electronic Circuit and System Design Springer Cham. p. 497-535, https://doi.org/10.1007/978-3-031-71436-8_14, (**2025)**.

[7] T. M. Hagan, D. B., Howard, B. Mark Hudson, J. D. Orlando, Neural Network Design, Second Edition, Martin Hagan, **(2014).**

[8] Y. Wang, Artificial-Intelligence integrated circuits: comparison of GPU, FPGA and ASIC, 3rd International Conference on Signal Processing and Machine Learning, 4: 99-104, https://doi.org/10.54254/2755-2721/4/20230358, **(2023).**

[9] S. M. Khan, A. Mann, AI Chips: What They Are and Why They Matter An AI Chips Reference, Center for Security and Emerging Technology (CSET), **(2020).**

[10] G.R. Sinha, Introduction to Sensors, Advances in Modern Sensors Physics, design, simulation and applications, IOP Publishing, **(2020).**

[11] R. Moheimani, et al., Recent Advances on Capacitive Proximity Sensors: From Design and Materials to Creative Applications, C, 8.2: 26, https://doi.org/10.3390/c8020026, **(2022).**

[12] R. Zou, et al., Progress of proximity sensors for potential applications in eletronic skins, Transactions of Tianjin University, 30.1: 40-62, https://doi.org/10.1007/s12209-023-00379-6, **(2024).**

[13] F. Noble, M. Xu, F. Alam, Static Hand Gesture Recognition Using Capacitive

Sensing and Machine Learning, Image Processing and Computer Vision in Smart Living Applications: Sensors, 23.7: 3419, https://doi.org/10.3390/s23073419, **(2023).**

[14] IBS Technical Resources, Comparing Capacitive and Inductive Sensors, **(2021).**

[15] B. Osoinach, Proximity Capacitive Sensor Technology for Touch Sensing Applications, Freescale Semiconductor, Proximity Sensing White Paper, **(2008).**

[16], M. Alshawabkeh et al., Highly Stretchable Additevly Manufactured Capacitive Proximity and Tactile Sensors for Soft Robotic Systems, IEEE Transactions on Instrumentation and Measurement, 72, 1-10, doi: 10.1109/TIM.2023.3250232, **(2023).**

[17] Z. Liu et al., A Self-Capacitance Proximity E-Skin With Long Range Sensibility for Robotic Arm Environment Perception, IEEE Sensors Journal, 23.13, 14854-14863, doi: 10.1109/JSEN.2023.3275206**, (2023).**

[18] K. Arimatus, H. Mori, Evaluation of Machine Learning Techniques for Hand Pose Estimation on Handheld Device with Proximity Sensor, CHI'20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, p. 1-13, https://doi.org/10.1145/3313831.337671, **(2020).**

[19] M. Virone, et al., Dynamic hand gesture recognition using a stretchable multi-layer capacitive array, proximity sensing, and a SVM classifier, 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), p. 7183-7188, doi: 10.1109/IROS51168.2021.9636744, **(2021).**

[20] A. Gupta, V. K. Sehrawat, M. Khosla, FPGA Based Real Time Human Hand Gesture Recognition System, 2nd International Conference on Communication, Computing & Security [ICCCS-2012], 6: 98-107, https://doi.org/10.1016/j.protcy.2012.10.013**, (2012).**

[21] P. Manickam, et al., Artificial Intelligence (AI) and internet of medical things (IoMT) assisted biomedical systems for intelligent healthcare, Biosensors, 12.8:562, https://doi.org/10.3390/bios12080562, **(2022).**

[22] K. N. Maanyu, et al., A Study on Tesla Autopilot, International Journal of Scientific Progress and Research (IJSPR), 71.01: 18-23, **(2020).**

[23] T. Szandala, Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks, Bio-inspired neurocomputing, Springer Singapore, Vol. 903 p.203-224, **(2020).** https://doi.org/10.1007/978-981-15-5495-7_11

[24] R. Mahima, et al., A Comparative Analysis of the Most Commonly Used Activation Functions in Deep Neural Network, 4th International Conference on Electronics and Sustainable Communication Systems, p. 1334-1339, doi: 10.1109/ICESC57686.2023.10193390, (**2023).**

[25] H. Choi, Y. Jang, Introduction to Neural Networks: DNN / CNN / RNN, EE807: Recent Advances in Deep Learning Lecture 1, KAIST EE Lecture Notes, https://alinlab.kaist.ac.kr/resource/Lec1_Introduction_to_NN.pdf, (Erişim Tarihi: **30 Mayıs 2025**).

[26] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, **(2016).**

[27] D. Stathakis, How many hidden layers and nodes? International Journal of Remote Sensing, 30.8: 2133-2147, https://doi.org/10.1080/01431160802549278, **(2009).**

[28] D. S. M. Maria, et al., Real-Time FPGA-Based Image Classification wtih Deep Neural Networks, 2024 RAEEUCCI, p. 1-7, doi: 10.1109/RAEEUCCI61380.2024.10547778**, (2024).**

[29] B. Li., J. Gu, W. Jiang Artificial Intelligence (AI) Chip Technology Review, 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), p. 114-117, doi: 10.1109/MLBDBI48998.2019.00028, **(2019).**

[30] Y. Chen, et al., Cloud-DNN: An open framework for mapping DNN models to Cloud FPGAs, Proceedings of the 2019 ACM/SIGDA international symposium on field-programmable gate arrays, p. 73-82, https://doi.org/10.1145/3289602.329391, (**2019).**

[31] P. D. Deotale, L. Dole, Design of FPGA based general purpose neural network, International Conference on Information Communication and Embedded Systems, p. 1-5, doi: 10.1109/ICICES.2014.7033843**, (2014).**

[32] R. Wu, et al., Accelerating Neural Network Inference on FPGA-Based Platforms – A Survey, MDPI Journals Electronics 10.9: 1025, https://doi.org/10.3390/electronics10091025, **(2021).**

[33] S. K. Venkataramanaiah, et al., Deep Neural Network Training Accelerator Designs in ASIC and FPGA, 2020 International SoC Design Conference, p. 21-22, doi: 10.1109/ISOCC50952.2020.9333063, **(2020).**

[34] P. P. Chu, RTL Hardware Design Using VHDL: Coding for Efficiency, Portability and Scalability, Wiley Interscience, **(2006).**

[35] V. Taraate, ASIC Design and Synthesis: RTL Design Using Verilog, Springer, **(2021).**

[36] K. Vipin, S. A. Fahmy, FPGA Dynamic and Partical Reconfiguration: A Survey of Architectures, Methods, and Applications, ACM Computing Surveys, Volume:51, Issue:4, Article No.:72, https://doi.org/10.1145/319382, **(2018).**

[37] N.H.E. Weste, D.M. Harris, CMOS VLSI Design A Circuits and Systems Perspective, Fourth Edition, Pearson, **(2010).**

[38] R. Arya, Hand Gesture Recognition Dataset, https://www.kaggle.com/datasets/aryarishabh/hand-gesture-recognition-dataset, (Erişim Tarihi: **01 Mayıs 2025**).

[39] GTI ve T. Mantecon, Hand Gesture Recognition Database, https://www.kaggle.com/datasets/gti-upm/leapgestrecog, (Erişim Tarihi: **01 Mayıs 2025**)

[40] A. Khan, Sign Language Gesture Images Dataset, https://www.kaggle.com/datasets/ahmedkhanak1995/sign-language-gesture-images-dataset, (Erişim Tarihi: **01 Mayıs 2025**).

[41] A. Akhouri, Hand Gestures Black And White, https://www.kaggle.com/datasets/flintytub49/hand-gestures-black-and-white, (Erişim Tarihi: **01 Mayıs 2025**).

[42] A. Singh, hand-sign-images, https://www.kaggle.com/datasets/ash2703/handsignimages, (Erişim Tarihi: **01 Mayıs 2025**).

[43] E. M. Erkut, Hacettepe University Computer Science CMP 784: Lecture Notes Fall 2023, https://web.cs.hacettepe.edu.tr/~erkut/cmp784.f23/lectures.html, (Erişim Tarihi: **02 Mayıs 2025**).

[44] C. Peng, et al., To What Extent Does Downsampling, Compression, and Data Scarcity Impact Renal Image Analysis, 2019 Digital Image Computing: Techniques and Applications (DICTA), p. 1-8, doi: 10.1109/DICTA47822.2019.8945813, **(2019).**

[45] R. C. Gonzales., R. E. Woods, Digital Image Processing 4th Edition, Pearson, **(2018).**

[46] A. Althbaity, M.M. Dessouky, I.R. Khan, Colorization of Grayscale Images Using Deep Learning, 14th IEEE International Conference on Computational Intelligence and Communcations Network, p. 131-138, doi: 10.1109/CICN56167.2022.10008319, **(2022).**

[47] J. Heaton, Introduction To Neural Networks For Java, Heaton Research Inc, 2005.

[48] Y. Li, C. Wei, T. Ma, Towards explaining the regularization effect of initial large learning rate in training neural networks, Advances in neural information processing systems NeurIPS 32, https://doi.org/10.48550/arXiv.1907.04595, **(2019).**

[49] J. M. Ede, R. Beanland, Adaptive Learning Rate Clipping Stabilizes Learning, Machine Learning: Science and Technology, IOP Publishing Ltd., 1.1: 015011, doi: 10.1088/2632-2153/ab81e2, **(2019).**

[50] M. Sokolova, N. Japkowicz, S. Szpakowicz, Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation, In: Sattar, A., Kang, Bh.

(eds), Advances in Artifical Intelligence Vol. 4304:1015-1021, pp.3, https://doi.org/10.1007/11941439_114, **(2006).**

[51] J. Chi, et al., Performance Analysis of Three Kinds of Neural Networks in the Classification of Mask Images, Journal of Physics: Conference Series. Vol 2181. No.1, doi: 10.1088/1742-6596/2181/1/012032**, (2022).**

[52] G. Szwoch, Gdańsk University of Technology, Department of Multimedia Systems, Application of signal processors Lecture Notes: Number Systems, https://sound.eti.pg.gda.pl/student/zps/en/04-Number.pdf, (Erişim Tarihi: **03.05.2025**).

[53] S.G. Aydin, H.Ş. Bilge, FPGA based implementation of sigmoid function using different, 2020 28h Signal Processing and Communication Applications Conference, p. 1-4, doi: 10.1109/SIU49456.2020.9302106, **(2020).**

[54] R. Pogiri, S. Ari, K. K. Mahapatra, Design and FPGA Implementation of the LUT based Sigmoid Function for DNN Applications, 2022 IEEE International Symposium on Smart Electronic Systems (ISES), p. 410-413, doi: 10.1109/iSES54909.2022.00090, **(2023).**

[55] F. D. Franco, et al., An Hardware Implementation of the Softmax Function on FPGA, CEUR-WS Volume:2768, **(2020).**

[56] X. Dong, X. Zhu, D. Ma, Hardware Implementation of Softmax Function Based on Piecewise LUT, 2019 IEEE International Workshop on Future Computing, p. 1-3, **(2019).**

[57] K. Vipin, Zynet: Automating Deep Neural Network Implemetation on Low-Cost Reconfigurable Edge Computing Platforms, 2019 International Conference on Field-Programmable Technology, p. 323-326, doi: 10.1109/IWOFC48002.2019.9078446, **(2019).**

[58] A. LeNail, Publication ready NN-architecture schematics, https://alexlenail.me/NN-SVG/, (Erişim Tarihi: **10 Haziran 2025**).

[59] A. Safanova et al., Ten deep learning techniques to address small data problems with remote sensing, International Journal of Applied Earth Observation and Geoinformation Volume 125: 103569, https://doi.org/10.1016/j.jag.2023.103569, **(2023).**

[60] A.H. Guedes et al., Performance Evaluation of Deep Learning Models For Image Classification Over Small Datasets: Diabetic Foot Case Study, IEEE Access 10: 124373-123386, doi: 10.1109/ACCESS.2022.3225107**, (2022).**

[61] I. Ahmad, S. Shin, An Approach to Run Pre-Trained Deep Learning Models on Grayscale Images, 2021 International Conference on Artificial Intelligence In

Information and Communication (ICAIIC), p. 177-180, doi: 10.1109/ICAIIC51459.2021.9415275, **(2021).**

[62] O. K. Oyedotun, K. Papadopoulos, D. Aouada, A New Perspective for Understanding Generalization Gap of Deep Neural Networks Trained with Large Batch Sizes, Applied Intelligence 53.12: 15621-15637, https://doi.org/10.1007/s10489-022-04230-8, **(2022).**

[63] X. Ying, An Overview of Overfitting and its Solutions, Journal of Physics: Conference Series Vol. 1168:2, doi: 10.1088/1742-6596/1168/2/022022, **(2019).**

[64] H. Bichri, A. Chergui, M. Hain, Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets, International Journal of Advanced Computer Science and Applications, Vol.15, No.2, doi: 10.14569/IJACSA.2024.0150235, **(2024).**

[65] V. R. Joseph, Optimal ratio for data splitting, Statistical Analysis And Data Mining The ASA Data Science Journal pp. 531-538, https://doi.org/10.1002/sam.11583, **(2022).**

[66] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, 2018 International interdisciplinary PhD workshop IEEE (IIPhDW), doi: 10.1109/IIPHDW.2018.8388338, **(2018).**

[67] T. Zhong, et al., RINAS: Training with Dataset Shuffling Can Be General and Fast, arXiv preprint, https://doi.org/10.48550/arXiv.2312.02368, **(2023).**

[68] K. Mukherjee, A. Khare, A. Verma, A Simple Dynamic Learning Rate Tuning Algorithm For Automated Training of DNNs, arXiv preprint, https://doi.org/10.48550/arXiv.1910.11605, **(2019).**

[69] D.H. Le, et al., Design of a Low-power Fixed-point 16-bit Digital Signal Processor Using 65nm SOTB Process, The International Conference on Integrated Circuit Design and Technology (ICICDT), p. 1-4, doi: 10.1109/ICICDT.2015.7165918, **(2015).**

[70] AMD/Xilinx, Zynq-7000 SoC Product Selection Guide (XMP097), pp.2, **(2019).**

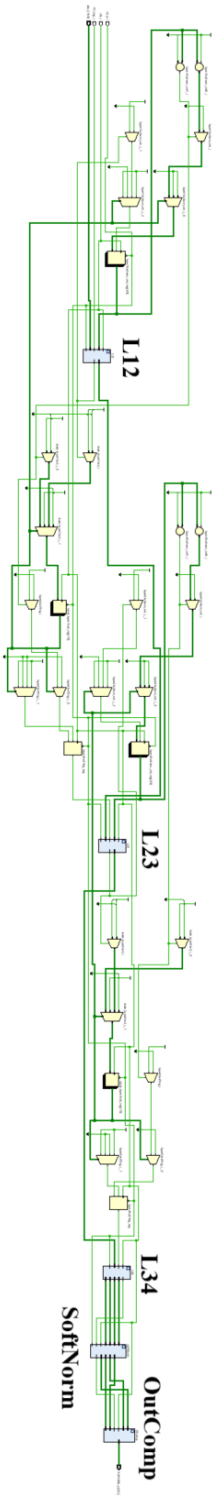
[71] C. H. Shen, et al., High-Speed-Recognition Artificial Intelligence Chip Based on ARM+FPGA Platform, 2024 IEEE 6th Eurasia Conference on IoT, Engineering Proceedings 92.1: 33, https://doi.org/10.3390/engproc2025092033**, (2025).**

[72] Xilinx, DSP: Designing for Optimal Results High-Performance DSP Using Virtex-4 FPGAs, Edition 1.0, **(2005).**

[73] D. Gökcen, Hacettepe Üniversitesi Elektrik-Elektronik Mühendisliği ELE719 Özel Konular (ASIC Tasarım) Ders Notları, **(2024).**

[74] Cadence, Genus Synthesis Solution: Massively parallel RTL synthesis and physical synthesis, Datasheet, **(2025).**
[75] Y. Wang, Artificial-Intelligence integrated circuits: comparison of GPU, FPGA and ASIC, 3rd International Conference on Signal Processing and Machine Learning, 4: 99-104, https://doi.org/10.54254/2755-2721/4/20230358, **(2023).**
[76] Cadence, Innovus Implementation System: Meet PPA and TAT targets at advanced nodes, Datasheet, **(2025).**

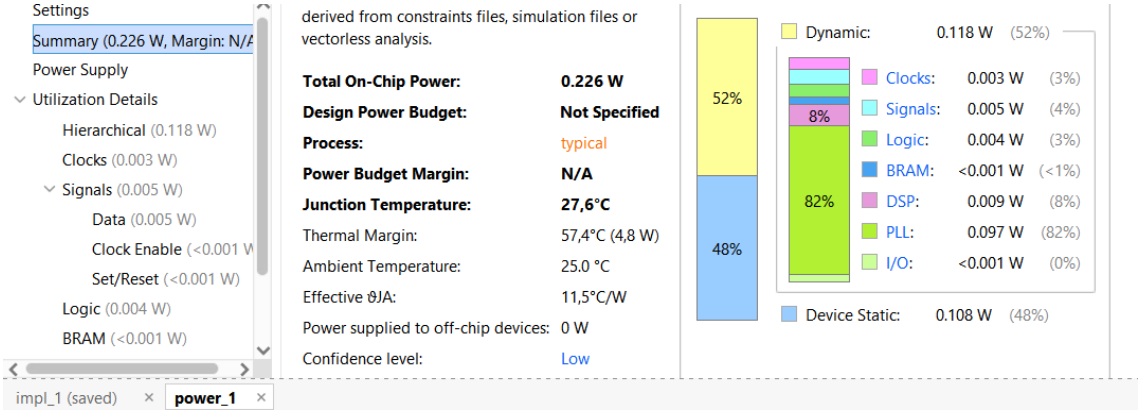
EK 1- Vivado yazılımı ile sayısal tasarım sonuçlarının incelenmesi



Şekil Ek.1. MLP Modelinin RTL Blok Diyagramı

Vivado 2023.2 yazılımı kullanarak MLP modelinin RTL blok diyagramı görünümü oluşturulmuştur. Katmanlar arasındaki veri transferini denetleyen modüller, blok

diyagramda isimlendirilerek belirtilmiştir. Şekil 3.13 incelendiğinde, 2. katman RTL'de "*L12*", 3. katman *"L23"* ve 4. katman yani çıktı katmanı ise *"L34"* olarak isimlendirilmiştir. Çıktı katmanda bulunan her bir düğümün çıkışı, Şekil 3.13'te bulunan karşılaştırıcıya yani *"OutComp"* tasarımına sürülerek işlem sonlanmaktadır. RTL blok diyagramında, *"SoftNorm"* bütün softmax fonksiyonu çıktılarını normalize eder. Şekil 5.11 incelendiğinde ASIC seviyesinde de bu tasarımın ayrı bir tasarım olduğu ve fazla sayıda mantıksal eleman kullandığı görülmektedir.



Şekil Ek.2. MLP Modelinin Güç Tüketim Raporu

Vivado yazılımı kullanılarak FPGA'da gerçekleştirilen sayısal tasarımın, güç analiz raporu oluşturulmuştur. Sayısal tasarımın etkili zaman analizi için PLL kullanılmıştır. FPGA ve ASIC'in güç tüketimi karşılaştırılırken, sadece MLP modelinin güç tüketimi karşılaştırmasını yapmak için PLL güç tüketimine dahil edilmemiştir. Güç raporu sonucunda elde edilen önemli parametreler Çizelge 4.8 incelenerek takip edilebilir.

EK 2 - Tez Çalışması Orjinallik Raporu

Şablona uygun olarak hazırlanan “Orijinallik Raporu”nun imzalı hali bu bölümde verilmelidir.