Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 35(2), ss. 477-483, Haziran 2020
Çukurova University Journal of the Faculty of Engineering and Architecture, 35(2), pp. 477-483, June 2020

Evrişimli Sinir Ağları için Maksimum Ortaklama Devre Tasarımları

Büşra BÜLBÜL[*1], Mustafa GÖK[1]

[1]*Çukurova Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Adana*

Geliş tarihi: 15.10.2019 Kabul tarihi: 30.07.2020

Öz

Derin Öğrenme uygulamaları hızla gelişmekte özellikle de mobil cihazlarda yaygın olarak kullanılmaktadır. Bu platformlardaki mevcut performans, güç ve alan kısıtları, uygulamaya özgü donanım tasarımlarına ihtiyacı artırmaktadır. Görüntü işleme alanındaki en güncel yöntemlerden başlıcası Evrişimli Sinir Ağları'dır. Bu çalışmada gelişkin Evrişimli Sinir Ağı mimarilerinin önemli bir işlem bloğu olan maksimum ortaklama ünite tasarımları sunulmuştur. Maksimum-ortaklama katmanı Evrişimli Sinir Ağı tasarımlarının kritik gecikme yolunda olup, boru hatlı bir tümleşik devrenin ana çevrim hızını etki edebilecek önemdedir. Önerilen tasarımların toplam çerçeve işleme süreleri Standart Tasarıma göre çok daha kısadır. Önerilen tasarımlar farklı boru hatlı yapılara entegre edilebilecektir. Tasarımlar VHDL ile modellenmiş ve güncel bir FPGA platformu üzerinde sentezlenmiştir. Sentez sonuçları, önerilen tasarımların en hızlısının Standart Tasarımla karşılaştırıldığında 128x128'lik bir çerçeveyi yaklaşık 8,1 kat daha hızlı işlediğini göstermiştir.

Anahtar Kelimeler: Maksimum-ortaklama, Evrişimli sinir ağı, Sayısal tasarım

Max-Pooling Circuit Designs for Convolutional Neural Networks

Abstract

Deep Learning applications are rapidly developing, especially in mobile devices. Existing performance, power and space constraints on these platforms increase the need for application-specific hardware designs. One of the most current methods in image processing is Convolutional Neural Networks. In this study, max-pooling unit designs, which is an important process block of Convolutional Neural Networks, are presented. The max-pooling layer is in the critical delay path of the Convolutional Neural Network design and is important to influence the main conversion rate of a pipeline integrated circuit. The total frame processing times of the proposed designs are much shorter than the Standard Design. The proposed designs can be integrated into different pipeline structures. All designs are modeled with VHDL and synthesized on a current FPGA platform. The synthesis results show that the fastest of the proposed designs processes a 128x128 frame around 8.1 times faster than the Standard Design.

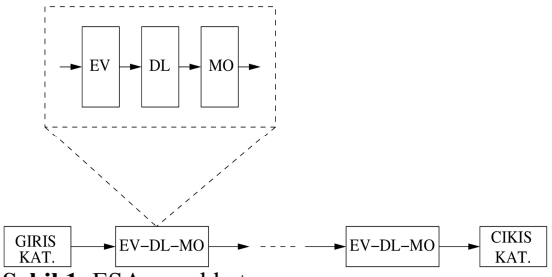
Keywords: Max-pooling, Convolutional neural network, Digital design

[*]Sorumlu yazar (Corresponding author): Büşra BÜLBÜL, *bbulbul@cu.edu.tr*

1. GİRİŞ

Derin öğrenme tabanlı uygulamalar modern hayatımızın vazgeçilmez bir unsuru haline gelmek üzeredir. Görüntü tanıma, ses tanıma, zaman dizisi analizi gibi alanlarda son yıllardaki ilerlemeler çok katmanlı derin ağların kullanılması sayesinde mümkün olmuştur [1]. Evrişimsel Sinir Ağı (ESA) modeli, derin öğrenme algoritmalarının yaygın olarak kullanılan bir çeşididir. Bu ağların işlem performansını artırmak için uygulamaya yönelik devre tasarımları son yıllarda oldukça ilgi çekmektedir [2-4]. Gerçek zamanlı sınıflandırma [5], hedef tespiti [6], bilgisayar destekli teşhis koyma [7], nesne algılama [8], konuşma tanıma [9], yüz tanıma [10] gibi pek çok alanda kullanılan donanım tasarımları yapılmaktadır.

ESA algoritmalarının çalıştırıldığı donanımlar kabaca üç sınıfa ayrılabilir. Birinci sınıfta Grafik İşlemci Tabanlı kartlar olup bunlar genelde algoritma tasarım ve ilk uygulama fazında kullanılan esnek donanımlardır. Grafik kartları kişisel bilgisayarlarda kullanılabileceği gibi bulut hizmetlerinin yığınlarına da eklenebilmektedir. Ancak, büyüklük ve güç tüketimi açısından ciddi dezavantajları vardır. Dolayısıyla bu donanımlar geliştirme aşamasında ve eğitim fazında tercih edilir. İkinci sınıfta FPGA tabanlı uygulamalar olup grafik tabanlı uygulamalara göre performans avantajına sahiptir. FPGA'lerin yeniden programlanabilir olmaları ve düşük miktarlarda üretimde maliyeti düşürmeleri gibi avantajları dolayısıyla tercih edilebilmektedir. FPGA tabanlı tasarımlar hem eğitim fazında hem de son ürün aşamasında kullanılabilir. Üçüncü sınıftaki donanımları Uygulamaya Yönelik İşlemciler sınıfına sokabiliriz. Bu donanımların eğitim fazından ziyade eğitilmiş ağların gerçeklenmesinde kullanımı uygundur. Yüksek performans isteminin maliyet kaygısına baskın olduğu alanlarda (askeri, sağlık vb.) tercih edilebilecek çözümlerdir. Son yıllarda cep telefonlarının işlemcileri ile bütünleşik çalışan yapay sinir ağı yongalarının da sisteme yerleştirilmesi yaygınlaşmaktadır. Bu donanımların büyük bir kısmı Uygulamaya Yönelik İşlemci sınıfındadır. Ancak, yığın üretimin daha düşük seviyede olduğu diğer alanlarda FPGA donanımlarına aktarılan fikri mülkiyet (Intellectual Property) modelleri satılmaktadır.

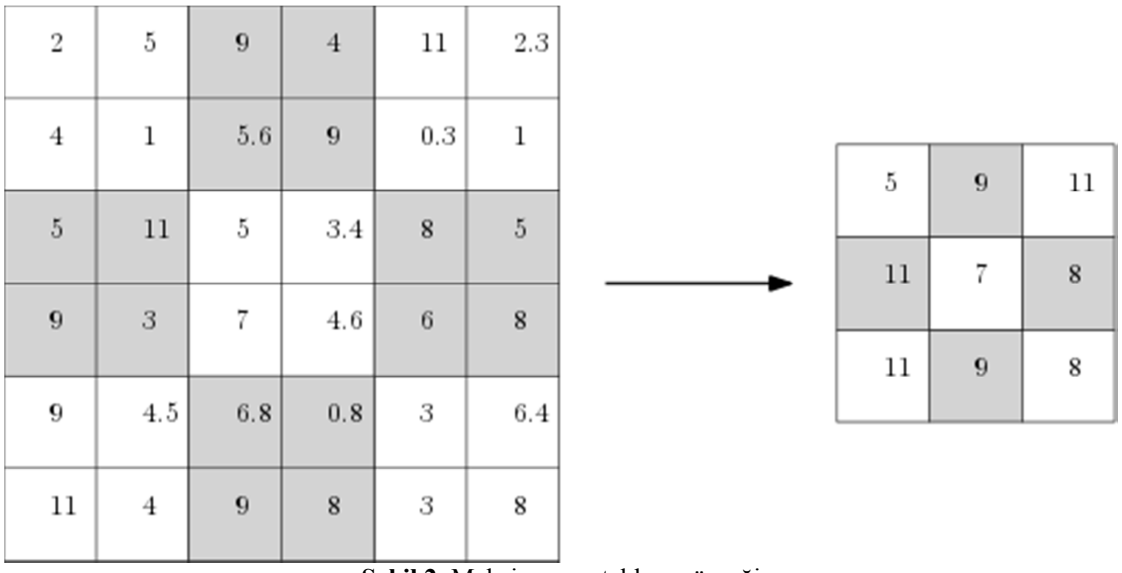


Şekil 1. ESA genel katman yapısı

Bu çalışmadaki tasarımlar FPGA platformlarına yerleştirileceği düşünülerek hazırlanmıştır. FPGA tabanlı uygulamalarda önemli olan sistem kaynaklarını dengeli kullanabilecek bir üst tasarım organizasyonu kurmaktır. Bu platformlarda performans kısıtı çoğu zaman boru hatlı bir organizasyon ile aşılmaya çalışılır. Bu nedenle tasarımlar şu anda üzerinde çalışılan boru hatlı çok katmanlı bir ESA tasarımına entegre olabilecek şekilde hazırlanmıştır.

2. EVRİŞİMLİ SİNİR AĞI YAPISI

ESA'nın giriş ve çıkış katmanları haricindeki derin yapısını oluşturan ara katmanlar genel olarak üç farklı işlem katmanın bir araya getirilmesi ile oluşmaktadır. Bu iç katmanları sırasıyla Evrişim katmanı (EV), Doğrultulmuş Lineer (DL) katman ve Maksimum Ortaklama (MO) olarak adlandırabiliriz. Şekil 1'de verilen blok şemada bu katmanların temsili sıralanması gösterilmiştir. Görüldüğü üzeri işlem blokları kolayca boru hattı bir donanım organizasyonu ile gerçeklenebilir. Boru hatlı bu yapıda EV en yoğun işlem yapılan katman olup genelde tasarım araştırmaları bu katmana yoğunlaşmıştır. DL katmanının görevi negatif değerlikli verileri sıfırla değiştirmek olduğu için minimum seviyede donanım kullanarak gerçekleştirilebilir. Çoğu zaman bu katman EV ile birleştirilir. Bu çalışmada MO katmanını gerçekleştirmeye yönelik devre tasarımları sunulmuştur. EV'nin de kendi içinde birkaç seviyeli boru hattına ayrılması durumunda MO performansı genel saat çevrim hızını etkileyecektir.



Şekil 2. Maksimum ortaklama örneği

3. MAKSİMUM-ORTAKLAMA İŞLEMİ

Alt-örnekleme katmanı olarak da bilinen ortaklama katmanının amacı işleme giren çerçeve boyutunu azaltmaktır. Bu işlem, maksimum ya da ortalama ortaklama yöntemlerini kullanarak gerçekleştirilebilir. MO katmanına giren veriler EV-DL katmanından aktarılan sabit noktalı pozitif verilerdir. Şekil 2'de ortaklama işlemine giren 6x6'lık bir çerçeve örnek olarak gösterilmiştir. Örnekte ortaklama 2x2'lik pencereler üzerinde yapılmaktadır. Her adımda pencere içindeki dört değer karşılaştırılmaktadır. Ortaklama çoğu zaman çakışmayan pencereler üzerinde yapılır. Bunu sağlamak için aşma miktarı (stride) 2 olarak belirlenir. Örnekte birbirinden bağımsız dokuz pencere vardır ve dokuz pencerenin maksimum değerlerinden oluşan yeni pencere sağ tarafta gösterilmektedir. Bu örnekteki değerler EV-DL katmanından çıkmıştır dolayısıyla tüm değerler pozitif sabit noktalı sayılardır.

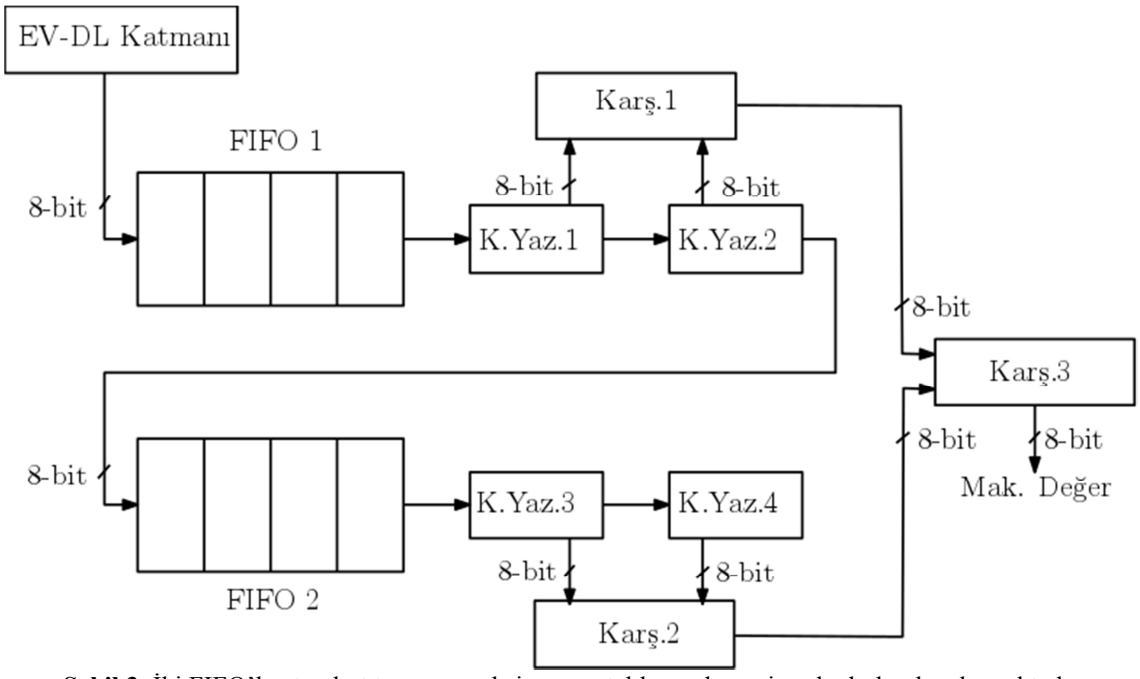
2x2'lik pencere en sık kullanılan olmakla beraber 3x3'lük pencere kullanan uygulamalarda mevcuttur. 3x3'lük pencere de aşma miktarı genelde 1 tutulur. Bu pencerelerin birbiriyle çakışmasına neden olur. Maksimum ortaklama dışında ortalama-ortaklama olarak adlandırılan ve çerçeve içindeki değerlerin ortalamasını alan işlem de vardır. Ancak, birtakım dezavantajları nedeniyle bu işlem güncel uygulamalarda terk edilmiştir.

3.1. Standart Tasarım

Şekil 3'de en sık kullanılan 2x2 maksimum ortaklama devre tasarımının blok şeması gösterilmiştir. İşlenen çerçevenin mxm boyutunda olduğu kabul edilirse FIFO'ların büyüklükleri m-2 bayttır. Bu devrede ayrıca dört adet kaydırıcı-yazmaç ve iki adet karşılaştırma devresi mevcuttur. Veri işleme akışının adımları şu şekildedir:

1) Birinci ve ikinci FIFO ve yazmaçlara EV-DL katmanından gelen çerçeve verilerinin birinci ve ikinci satırı 2 m çevrimde doldurulur.

2) Sonraki adımda her çevrimde FIFO'dan okunan değerler önce ikişer ikişer Karşılaştırıcı 1 ve 2'de karşılaştırılır. Bulunan büyük değerler Karşılaştırıcı 3'de son defa kıyaslanarak penceredeki maksimum değer bulunur.



Şekil 3. İki FIFO'lu standart tasarım maksimum ortaklama devresi: ardışıl olarak çalışmaktadır

Standart tasarım m x m'lik bir penceredeki tüm maksimumları (3 m^2+m)/2 çevrimde bulur. FIFO'lar dolduktan sonra her iki çevrimde bir 2x2'lik yeni pencerenin maksimum değeri bulunur. Önceki çalışmalarda kullanılan standart maksimum ortaklama donanımları genelde bu yapıdadır.

4. TASARIMLAR

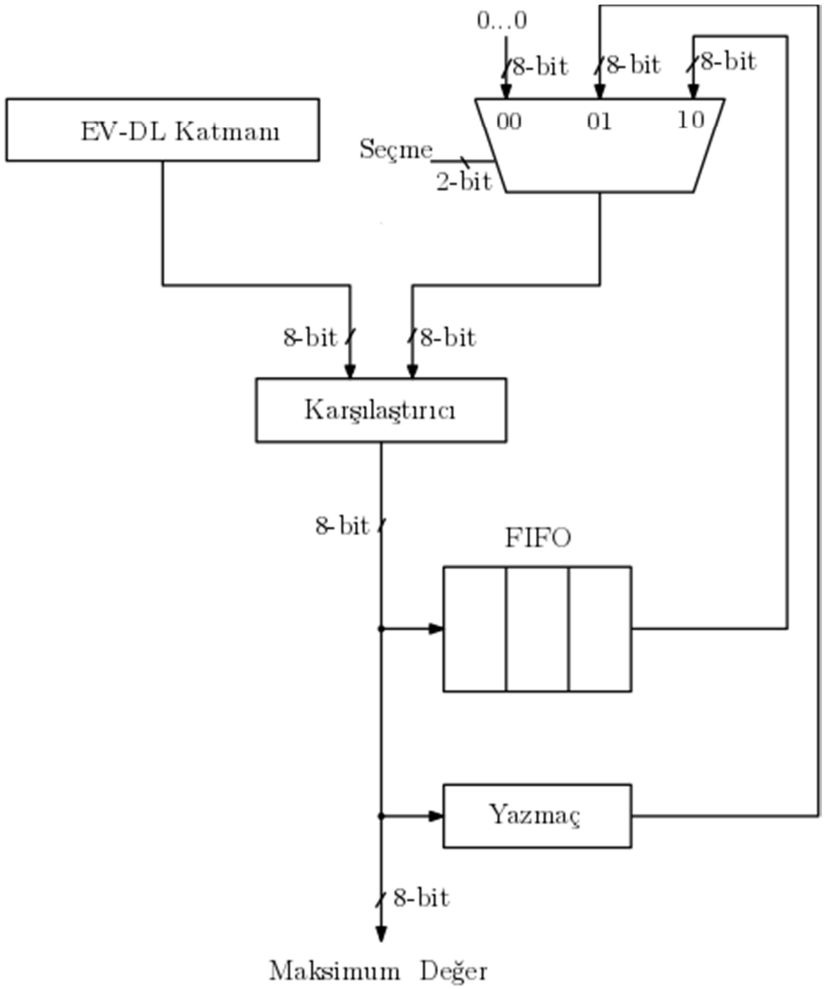
Bu kısımda farklı iki boru hatlı organizasyonuna uygun 2x2'lik pencere ve aşma miktarı 2 olan tasarımlar anlatılmaktadır. MO katmanında bu tasarımların birden fazla kopyasının aynı anda işlem yapacağı farz edilmiştir.

4.1. Tasarım 1

Birinci MO tasarımı her saat çevriminde EV-DL katmanından bir adet değerin gönderildiği kabul edilerek hazırlanmıştır. Şekil 4'de blok şeması gösterilen ünitede bir adet 8-bitlik yazmaç, bir adet karşılaştırıcı, bir adet multipleksır, bir adet FIFO kullanılmıştır. İşlenen çerçevenin mxm boyutunda olduğunu kabul edilirse FIFO'nun büyüklüğü m/2 bayt kadardır. Çerçevenin tamamı işlendikten sonra çerçevenin boyutu m/2 x m/2'ye düşecektir. Sürecin ana adımları şu şekilde gerçekleşir:

1) Girişteki değer genlik karşılaştırıcıya gönderilir. Tek sayılı çevrimlerde multipleksırın '00' girişi seçilir; gelen değer yazmaca kaydedilir; FIFO'ya yazma izni verilmez.
2) Bir sonraki çevrimde multipleksırın '01' girişi seçilir. Girişe gelen ikinci değer yazmaçta bulunan değerle karşılaştırılır; büyük olan FIFO'ya yazılır. Bir önceki adımla bu adım çerçevenin bir satırı bitene kadar sürdürülür. 1. ve 2. adımlar m çevrim süresince tekrarlanır.
3) Yeni satıra geçildiğinde; tek sayılı çevrimlerde multipleksırın '10' girişi seçilir, bu durumda yeni gelen değerler FIFO'dakilerle karşılaştırılır. Çift sayılı çevrimlerde ise multipleksırın '01' girişi seçilir ve yeni gelen değerler yazmaçtaki değerlerle karşılaştırılır. Karşılaştırıcının çıkışı her çevrimde yazmaca yazılır. Bu adım da m çevrim sürdürülür.

Özetle tasarım tek sayılı satırlardan gelen değerleri ikişer ikişer karşılaştırıp FIFO'ya kaydeder. Çift sayılı satırlardan gelen değerler ise FIFO'daki değerlerle karşılaştırılır. Tasarım 1, FIFO dolduktan sonra her iki çevrimde bir maksimum değer bulur. Bu nedenle mxm'lik bir çerçeve üzerinde MO işlemi toplam m^2 çevrimde gerçekleşir.



Şekil 4. Tek FIFO'lu maksimum ortaklama tasarımı: mxm'lik pencere işleme hızı m^2

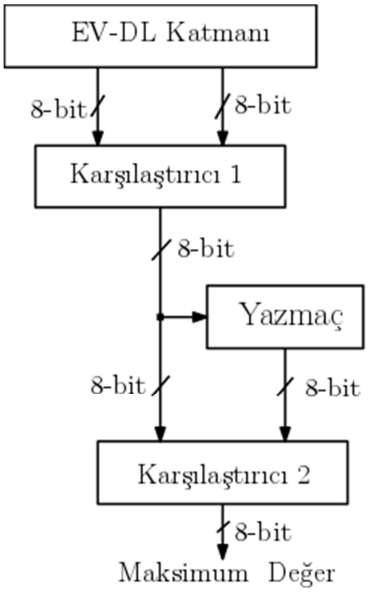
4.2. Tasarım 2

Tasarım 2, EV-DL katmanında çerçevenin iki satırını aynı anda işleyen donanımın kullanılması durumunda tercih edilebilecektir. Şekil 5'de blok şeması gösterilen ünitede bir adet yazmaç, iki adet karşılaştırıcı kullanılmıştır. Bir önceki tasarımda kullanılan FIFO'ya ve multipleksıra ihtiyaç duyulmamıştır. Bu tasarımın çalışmasının veri akışının ana adımları aşağıda verilmiştir.

1) İki değer karşılaştırıcıda karşılaştırılır ve büyük olan yazmaca kayıt edilir.

2) İkinci çevrimde gelen iki değerin büyüğü bulunduktan sonra yazmaçtaki değerle ikinci karşılaştırıcı tarafından karşılaştırılır. Çift sayılı çevrimlerde yazmaca yeni değer girişine izin verilmez.

Tasarım 2'de her iki çevrimde mxm'lik bir çerçeve üzerinde MO işlemini toplam $m^2/2$ çevrimde gerçekleştirir.



Şekil 5. İki veri yollu maksimum ortaklama devresi: mxm'lik pencere işleme hızı $m^2/2$

5. SONUÇLAR

Bu çalışmada sunulan maksimum-ortaklama devre tasarımları VHDL ile modellenmiştir. Sentezlenen modeller 128x128'lik bir pencereyi işleyecek boyutta üretilmiştir. Pencerenin boyutu FIFO büyüklüğünü doğrudan etkileyici bir faktör olduğu için boyut özellikle belirtilmiştir. Tasarım FIFO boyutu değiştirilerek daha küçük veya daha büyük pencereleri işleyecek şekilde ölçeklenebilir. Penceredeki her bir piksel 8 bitlik sabit noktalı sayı olarak kabul edilmiştir. Her tasarım için karşılaştırma devrelerinde çıkartma ünitesi veya genlik karşılaştırma ünitesi kullanılmıştır. Bunun nedeni karşılaştırma devrelerinin kritik gecikme yolu üzerinde olmalarıdır. Nitekim genlik karşılaştırma devreleri ile sentezlenen tasarımlar daha yüksek saat hızı sonuçları vermiştir. Her tasarımın iki versiyonu da fonksiyonel olarak doğrulandıktan sonra Quartus II yazılımı ile Intel Cyclone V: 5CGXFC7C7F23C8 FPGA'sı üzerine aktarılmıştır. Çizelge 1'de FPGA donanım kaynaklarının kullanımı verileri sunulmuştur. Bu veriler çizelgedeki 1. sütunda sırasıyla; uyarlanabilir mantık modülü, bilişimli uyarlanabilir aramalı tablo, atanmış mantık yazmaçları, giriş/çıkış pini, maksimum çıkış yelpazesi, toplam çıkış yelpazesi ve ortalama çıkış yelpazesi olmak üzere yedi başlık altında verilmiştir. Bu çizelgede 2., 3., 5. ve 6. sütunlarda önerilen tasarımların sonuçları, 4. ve 7. sütunlarda standart tasarımın sonuçları gösterilmektedir. Standart Tasarım tüm başlıklarda daha fazla FPGA kaynağı kullanmaktadır. Standart Tasarımın en önemli dezavantajı ihtiyaç duyduğu FIFO büyüklüğünün fazla olmasıdır. Tasarım 2 donanım kaynakları kullanımı bakımından tüm tasarımlardan daha iyi durumdadır. Bunun başlıca sebebi FIFO kullanmaması ve düzenli bir bağlantı yapısına sahip olmasındandır. Tasarım 2'nin EV-DL katmanından çift veri yolu ile beslenmesi gerektiği unutulmamalıdır.

Çizelge 1. FPGA donanım kaynaklarının kullanımı

Kaynak	Genlik-karş. kullananlar			Çıkartma dev. kullananlar		
	T1	T2	ST	T1	T2	ST
Mantık modülü	74	20	129	70	19	125
LUT	97	28	172	100	37	178
Yazm.	59	10	138	59	10	138
G/Ç pini	18	26	18	18	26	18
Maks. çıkış	67	10	154	67	17	154
Toplam çıkış	734	193	1426	705	192	1415
Ort. çıkış	3,67	2,14	3,94	3,47	1,94	3,85

Çizelge 2'de ise 85C ve 0C modelleri üzerinde hesaplanan maksimum frekans sonuçları verilmiştir. Standart Tasarımın maksimum frekans sonucu Tasarım 1'den daha hızlı Tasarım 2'den daha yavaştır. Diğer yandan toplam çerçeve işlem zamanı da önemlidir. mxm'lik bir çerçeveyi Tasarım 1 toplam m^2 çevrimde, Tasarım 2 $m^2/2$ çevrimde, Standart Tasarım $(3\ m^2+m)/2$ çevrimde hesaplamaktadır. Bu göz önüne alınırsa her ne kadar Tasarım 1'in saat hızı Standart Tasarım'dan yavaşsa da çerçeve işlem süresi daha kısa olacaktır. Örneğin Tasarım 1, 128x128'lik çerçeveyi yaklaşık 141 µs'de hesaplarken, Standart

Tasarım 154 µs'de hesaplar. Bu çerçeveyi Tasarım 2 ise 19 µs'de hesaplar. Tasarım 2'nin üstünlüğü çift veri yolu ile beslenmesi ve daha kısa kritik-yol yapısına sahip olmasındandır.

Sonuç olarak Tasarım 2 kullanılması ESA'nın MO hesaplama aşamasında hızlandırılmasını sağlayacaktır. Tasarım 2'nin kullanılması için önceki işlem katmanlarının da ona uygun olarak tasarlanması gerekecektir.

Çizelge 2. FPGA maksimum saat çevrim sonuçları

Tasarım	Genlik-karş. kullananlar		Çıkartma dev. kullananlar	
	Slow 1100mV 85C (MHz)	Slow 1100mV 0C (MHz)	Slow 1100mV 85C (MHz)	Slow 1100mV 0C (MHz)
T1	115,89	114,09	114,78	111,20
T2	426,08	442,87	408,50	419,46
ST	160,03	161,92	150,22	150,15

6. KAYNAKLAR

1. Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
2. Hamdan, M.K., Rover, D.T., 2017. VHDL Generator for a High Performance Convolutional Neural Network FPGA-based Accelerator. In 2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig) 1-6. IEEE.
3. Dinelli, G., Meoni, G., Rapuano, E., Benelli, G., Fanucci, L., 2019. An FPGA-based Hardware Accelerator for CNNs Using On-chip Memories Only: Design and Benchmarking with Intel Movidius Neural Compute Stick. International Journal of Reconfigurable Computing. Hindawi.
4. Shawahna, A., Sait, S.M., El-Maleh, A., 2018. FPGA-based Accelerators of Deep Learning Networks for Learning and Classification: A Review, 7823-7859. IEEE Access.
5. Hwang, W.J., Jhang, Y.J., Tai, T.M., 2017. An Efficient FPGA-based Architecture for Convolutional Neural Networks. In 2017 40th International Conference on Telecommunications and Signal Processing (TSP), 582-588, IEEE.
6. Li, Y., Song, B., Kang, X., Du, X., Guizani, M. 2018. Vehicle-type Detection Based on Compressed Sensing and Deep Learning in Vehicular Networks. Sensors, 18(12), 4500.
7. Rajaraman, S., Candemir, S., Kim, I., Thoma, G., Antani, S., 2018. Visualization and Interpretation of Convolutional Neural Network Predictions in Detecting Pneumonia in Pediatric Chest Radiographs. Applied Sciences, 8(10), 1715.
8. Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards Real-time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems, 91-99.
9. Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Penn, G., 2012. Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4277-4280, IEEE.
10. Qiao, S., Ma, J., 2018. FPGA Implementation of Face Recognition System Based on Convolution Neural Network. In 2018 Chinese Automation Congress (CAC), 2430-2434, IEEE.