

YENİ NESİL ARM ÇEKİRDEKLERİN ÇOKLUORTAM ALGILAYICI AĞLAR İÇİN GÖRÜNTÜ İŞLEME YETENEKLERİ

IMAGE PROCESSING CAPABILITIES OF NEW GENERATION ARM CORES FOR WMSNs

N. Cihan Camgöz, Recep Öztürk, M. Amaç Güvensan, Z.Cihan Tayşi, A. Gökhan Yavuz

Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi

{ncihan, recep, amac, cihan, gokhan}@ce.yildiz.edu.tr

ÖZETÇE

Son yıllarda Telsiz Algılayıcı Ağlar konusunda hızlı gelişmeler yaşanmıştır. Özellikle yakın zamanda, CMOS kamera ve mikrofonlar gibi donanımların ucuzlaşarak yaygınlaşması Telsiz Çokluortam Algılayıcı Ağların (TCAA) gelişiminde önemli rol oynamıştır. Bu ağlarda çevreden toplanan bilginin yorumlanması için yüksek işlem gücüne sahip düğümlere ihtiyaç duyulmaktadır. TCAA'ların yapısında kullanılan düğümlerin büyük çoğunluğunda ARM7 çekirdeğine sahip mikrodenetleyiciler kullanılmaktadır. Son zamanlarda gömülü sistem uygulamalarında kullanılan uygun maliyetli ve esnek yapıdaki ARM9 ve ARM11 çekirdeklerine sahip mikrodenetleyiciler ARM7 çekirdeğine alternatif oluşturmaktadır. Bu çalışmada ARM9 ve ARM11 çekirdekleri işlem gücü ve enerji tüketimi açısından karşılaştırılmıştır. Gerçekleştirilen testler sonucunda ARM11 çekirdeğinin ARM9 çekirdeğine göre, işlemleri 3 ile 4 kat arasında daha hızlı gerçekleştirdiği gözlemlenmiştir.

ABSTRACT

Last decade witnessed the rapid development of Wireless Sensor Networks (WSNs). More recently, the availability of inexpensive hardware such as CMOS cameras and microphones that are able to ubiquitously capture multimedia content from the environment has fostered the development of Wireless Multimedia Sensor Networks (WMSNs). Nodes in such networks require significant amount of processing power to interpret the collected sensor data. Most of the currently available wireless multimedia sensor nodes are equipped with ARM7 core microcontrollers. On the other hand, ARM9 and ARM11 cores are viable alternatives, which deliver deterministic high performance and flexibility for demanding and cost-sensitive embedded applications. Thus, we evaluated the performance of the ARM9 and the ARM11 cores in terms of processing power and energy consumption. Our test results showed that the ARM11 core performed 3 to 4 times faster than the ARM9 core.

1. GİRİŞ

Telsiz Çokluortam Algılayıcı Ağlar, algılayıcı teknolojileri, gömülü bilgisayar sistemlerindeki ilerlemeler ve pahalı olmayan CMOS kamera ve algılayıcıların kolay erişilebilirliği sayesinde oldukça yaygınlaşmıştır [1]. TCAA'lar, bulunduğu ortamdan kısıtlı sayısal veriler toplayabilen geleneksel Telsiz Algılayıcı Ağların aksine, bulunduğu ortamdan görsel/işitsel bilgi elde etme yeteneğine sahiptir. Bunun sonucunda, ortam izleme, trafik kontrol, gelişmiş sağlık hizmetleri, endüstriyel işlem kontrolü vb. görevleri yerine getiren TCAA'lar yakın zamanda hayata geçirilmiştir [2].

TCAA'larda uygulamaya bağlı olarak genelde büyük miktarda veri işlenir [3]. Algılayıcı düğümlerin pil kısıtları, hat kapasitelerinin yetersizliği vb. nedenlerle sadece anlamlı bilgilerin baz istasyonuna transferi önem kazanmaktadır. Öte

Bu çalışma Y.T.Ü. BAPK tarafından desteklenmektedir

yandan, yeni nesil mikrodenetleyiciler, araştırmacıları, kodlama, sıkıştırma, nesne algılama vb. sinyal işleme algoritmalarını düğümlerde gerçekleştirmek konusunda teşvik etmektedir. Bu gelişmeler sonucunda, yerinde işleme (in-situ processing) TCAA'larda ön plana çıkmıştır.

Nesneler ve/veya bölgeler bir resim/çerçeveadaki en önemli anlamlı bilgi olarak tanımlanabilmektedir. Bir TCAA'da, düğümler, hedeflenen nesnelere algılama, tanıma ve takip etmekten sorumludurlar [3]. Bu görevleri yerine getirebilmek için düğümler üzerinde birçok görüntü işleme tekniği uygulanmaktadır. Mevcut düğümler üzerinde gerçek zamanlı görüntü işleme, özellikle de gerçek zamanlı video işleme, işlemci ve bellek kısıtlarından dolayı hala zordur. [4]'de, görüntü işleme gibi karmaşık işlemlerin gerçekleştirilmesinde yüksek işlem gücüne sahip mikrodenetleyicilerin kullanılmasının daha verimli olduğu ortaya konmuştur. Buna dayanarak yazarlar, bir Intel PXA255 mikrodenetleyici ve bir TI MSP430F1611 mikrodenetleyici içeren çift işlemcili bir mimari önermektedirler. PXA255 mikrodenetleyicisi sadece görüntü işleme işlemleri için kullanılırken, MSP430F1611 mikrodenetleyicisi genel algılayıcı işlemleri için kullanılmaktadır. Günümüzde kullanılan Telsiz Çokluortam Algılayıcı düğümlerinde [1][5] yaygın olarak düşük maliyetleri, küçük boyutları ve basit yapıları nedeniyle ARM mikrodenetleyicilerin kullanıldığı görülmektedir.

ARM [6], ARM7, ARM9 ve ARM11 gibi farklı işlevselliğe ve kabiliyete sahip birçok mikrodenetleyici çekirdeği seçeneği sunmaktadır. Yapılan testler [], ARM7'nin performansının özellikle yüksek çözünürlüklü gerçek zamanlı video işleme için gerekli seviyenin altında olduğunu göstermektedir. Öte yandan ARM9 ve ARM11 sinyal işleme algoritmaları için yeni beceriler sunmaktadır. Bununla birlikte ARM11 özellikle çokluortam uygulamaları için düzenlendiğinden LCD denetleyicisi, MPEG kodlayıcısı ve grafik hızlandırıcısı gibi geniş bir çevre birim yelpazesini bünyesinde barındırmaktadır. Ön işlemeden kodlamaya kadar pek çok görüntü/video işleme algoritması, özel matrislerden [7] faydalanarak görüntüyü dönüştüren, bir dizi konvolüsyondan oluşmaktadır. Bu çalışmada konvolüsyon işlemi özelinde, ARM9 ve ARM11'in performansları karşılaştırılarak, ARM11 tabanlı düğümlerin, TCAA'larda gerçekleştirilecek sinyal işleme uygulamalarına sağlayacağı avantajlar ortaya

konulmaktadır.

2. GÖRÜNTÜ İŞLEMEDE KONVOLÜSYON-TABANLI ALGORİTMALAR

Konvolüsyon, görüntü işleme için temel matematiksel bir işlemdir. Pek çok görüntü işleme algoritmasında konvolüsyon, görüntüyü iyileştirmek ve/veya görüntüye ait öznelik çıkarımı için kullanılmaktadır [3]. Bu işlemde, görüntü, boyutu ve şekli daha önceden tanımlanmış matris formunda bir pencere ile taranır. Bu pencere, konvolüsyon matrisi olarak adlandırılır ve pencerenin her bir hücresi uygulanacak filtreye ait ağırlık değerlerini içerir. Matrisin merkezinde yer alan pikselin yeni değeri pencere içindeki pikseller ile filtre değerlerinin ağırlıklı toplamıdır. $F := (f(x, y)_{m \times n})$, $2B$ bir görüntü, H bir konvolüsyon matrisi ise, $g(x, y)$ yeni görüntüyü ifade etmektedir.

$$g(x, y) = f(x, y) \otimes h(x, y); \quad (1)$$

Bu işlem Eşitlik 2'deki gibi ifade edilebilir.

$$g(x, y) = \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} h[j+(n-1), i+(n-1)] \times f[x-j, y-i] \quad (2)$$

Konvolüsyon parazit temizleme, yumuşatma, netleştirme, keskinleştirme, kenar belirleme vb. pek çok farklı amaç için görüntülere uygulanmaktadır. Konvolüsyon tabanlı görüntü işleme algoritmaları üç temel gruba ayrılabilir:

1. İyileştirme İşlemleri
2. Türev-Tabanlı İşlemler
3. Morfolojik İşlemler

2.1. İyileştirme İşlemleri

Görüntü iyileştirme operasyonları, bölütleme ve özellik çıkarımı gibi ileri işlemler için görüntüyü hazırlamak amacıyla gerçekleştirilir. Bu tür filtreler parazit temizleme, yumuşatma, bulanıklaştırma ve keskinleştirme işlemlerinde kullanılmaktadır. Mevcut görüntü filtreleri, doğrusal ve doğrusal olmayan, düşük geçirgen ve yüksek geçirgen, dikdörtgenel ve dairesel gibi pek çok alt kümeye ayrılabilir. Örnek olarak, en yaygın kullanılan yüksek geçirgen ve alçak geçirgen filtreler Eşitlik 3'te verilmiştir.

$$L_{pass_F} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_{pass_F} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3)$$

2.1.1. Türev-Tabanlı İşlemler

Türev-tabanlı operasyonlar bir görüntüde üç temel tip devamsızlığı yakalamak için kullanılır; *nokta*, *çizgi* ve *kenar* [8]. İki tür uygulama vardır; birinci dereceden türev ve ikinci dereceden türev. En yaygın birinci dereceden türev-tabanlı

işlem kenar belirlemedir. İki boyutlu bir görüntüde kenarları belirlemek için, görüntü üzerinde bir yatay ve bir dikey pencere gezdirilir. Tarama sırasında, eğim yönü ve pikselin yeni değeri hesaplanır. Birinci dereceden türevler arasında en fazla tercih edilen gradyan matrisleri Roberts, Prewitt, Sobel ve Gaussian filtreleridir. Eşitlik 4'te Prewitt filtreleri sunulmaktadır.

$$P_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (4)$$

2.2. Morfolojik Operasyonlar

Görüntü işlemedeki morfolojik operasyonlar, sınırlar, iskeletler ve dış bükey kabuklar gibi bölgedeki şekillerin tanımlanması ve temsil edilmesinde çok yararlı olan görüntü bileşenlerini ayırtmayı hedeflemektedir. Morfolojik operasyonların dört temel türü vardır; *genişletme (dilation)*, *daraltma (erosion)*, *açma (opening)*, *kapama (closing)*. Genişletme nesnelerin boyutlarının büyümesine neden olurken, daraltma nesnelerin boyutlarının küçülmesine sebep olur. Genişletme ve daraltmanın miktarı ve şekli, kullanılan yapısal nesnenin, örneğin konvolüsyon matrisi, seçimine bağlıdır. Öte yandan açma ve kapama, genişletme ve daraltmanın kombinasyonlarıdır. Açma işleminde, görüntüye önce daraltma sonra genişletme uygulanır. Açma genellikle görüntünün dış çizgilerini yumuşatır, dar kanalları ve ince çıkıntıları ortadan kaldırır. Kapamada, daraltma ve genişletme açmadaki sıranın tam tersi şekilde uygulanır. Kapama dış hatları yumuşatmayı hedeflese de açmanın tersine, dar kırıkları ve uzun ince olukları genişletir, küçük delikleri eler ve dış hatlardaki boşlukları doldurur.

2.3. Konvolüsyon İşleminin Karmaşıklık Analizi

İki boyutlu konvolüsyon yoğun bir hesaplama gerektirir. $N \times N$ 'lik bir görüntü ve $M \times M$ 'lik bir matris için konvolüsyonun zaman karmaşıklığı $O(N^2 M^2)$ şeklindedir. Konvolüsyon işlemi, bir dizi yükleme-çarpma-atama (load-multiply-store) operasyonudur. Bu nedenle bir Çarpma ve Toplama (Multiply-and-Accumulate) operasyonu olarak değerlendirilebilir. Donanım olarak Çarpma ve Toplama desteğine sahip işlemciler konvolüsyon işlemi daha hızlı gerçekleştirebilirler. Konvolüsyonun sözde kodu aşağıdaki gibidir.

Algorithm 1 2B Konvolüsyon İşlemi için Sözde Kod

```

CONVOLUTION(*srcImg,width,height,convKernel,kernelSize,*dstImg)
for i = kernelSize/2 to height - (kernelSize/2) do
  for j = kernelSize/2 to width - (kernelSize/2) do
    tmp=0
    for k = -kernelSize/2 to kernelSize/2 do
      for l = -kernelSize/2 to kernelSize/2 do
        tmp=tmp+convKernel[k][l] × srcImg[i+k][j+l];
      end for
    end for
    dstImg[i][j]=tmp;
  end for
end for

```

3. ARM9 ve ARM11 KARŞILAŞTIRILMASI

Komut seti açısından ARM11 çekirdeği, ARM9 çekirdeği ile uyumlu olsa da, ARM11 çekirdeği ARMv6 mimarisi üzerine kurulmuştur. ARMv6 mimarisinin kullanılması ile ARM11 çekirdeği, daha yüksek çalışma frekansı, çokluortam işleme eklentileri, geliştirilmiş ön bellek mimarisi, geliştirilmiş işhattı ve 64-bit veri yolu kullanımı gibi avantajlar sunmaktadır.

3.1. Çalışma Frekansı

TCAA'lar, çevreden toplanan ve işlenmesi gereken bilginin büyüklüğünden dolayı yüksek işlem gücüne ihtiyaç duyarlar. Ancak tüm Telsiz Algılayıcı Ağlar için geçerli olan düşük güç tüketimi ve maliyet kısıtlarını da taşımaktadırlar.

ARM11 çekirdeğine sahip ilk mikrodenetleyiciler 350 – 500MHz, sonraki mikrodenetleyiciler ise 1GHz çalışma frekansına ulaşmıştır. ARM9 çekirdeğine sahip mikrodenetleyicilerin çalışma frekanslarının 200 – 533MHz olduğu düşünüldüğünde ARM11 çekirdeğine sahip mikrodenetleyicilerin çalışma frekansı bakımından iki kata kadar üstün oldukları görülmektedir. Ayrıca ARM11 çekirdeğine sahip mikrodenetleyicilerin, ARM9 çekirdeğine sahip mikrodenetleyiciler gibi çalışma frekansını dinamik olarak değiştirerek güç tüketimini dengelemeleri mümkündür.

3.2. Çokluortam İşleme Eklentileri

ARMv6 mimarisi ses ve video işlemeye yönelik çok sayıda komut içermektedir. Bu ek komutların kullanılması durumunda MPEG4 kodlama ve çözme işlemlerini önceki mimarilere göre iki kat daha hızlı gerçekleştirilebilmektedir. Ancak bu komutların uygulamaya özel olmaları nedeniyle TCAA uygulamalarında kullanılması her zaman mümkün değildir.

3.3. Geliştirilmiş Önbellek Mimarisi

ARMv6 mimarisinde önbellek fiziksel olarak adreslenmiştir. Bu şekilde önbellek yapısında yaşanan örtüşmelerin (aliasing) önüne geçilmiştir. Önbellek yapısındaki bu değişim, bağlam anahtarlama (context switch) işleminin de daha hızlı yapılmasını sağlamaktadır.

3.4. Geliştirilmiş İstisna ve Kesme Değerlendirmesi

ARMv6 mimarisinde 31 adet 32-bit yazmaç bulunmaktadır. Bu yazmaçlardan 15 tanesi çalışma sırasında oluşan istisnai durumların ve kesmelerin işlenmesinde kullanılmaktadır. Bu şekilde uygulamaların çalışmasında yaşanan kesintiler en aza indirilmektedir.

3.5. İşhattı Yapısı

ARMv6 mimarisinin temel tasarım hedefi yüksek işlem gücü ve düşük güç tüketimidir. Yüksek işlem gücünün sağlanabilmesi için işhattı yapısında çeşitli değişiklikler yapılmıştır. ARM9 çekirdeğinde beş olan kademe sayısı 8'e çıkartılmıştır. Uzun işhatlarında işlemler arası bağımlılık ve dallanma işlemleri gibi nedenlerden oluşabilecek gecikmeleri engellemek için *dallanma tahmini ve işlem ötelenmesi* teknikleri kullanılmaktadır.

Yapısal değişiklikler ve kullanılan teknikler sayesinde ARM11 çekirdeği, ARM9 çekirdeğine oranla 40% oranında daha yüksek işlem gücü sağlayabilmektedir.

3.6. 64-bit veri yolu

ARMv6 mimarisinde yapısında tamsayı işlem birimi ile komut ve veri önbellekleri arasında 64-bit uzunluğunda veri yolları kullanılmıştır. 64-bit uzunluğunda veri yollarının kullanılması bir çevrim içerisinde iki komutun önbellekten işlemciye aktarılmasına imkan vermektedir. Bu şekilde çekirdek performansında ciddi bir artış sağlanmaktadır.

4. PERFORMANS ANALİZİ

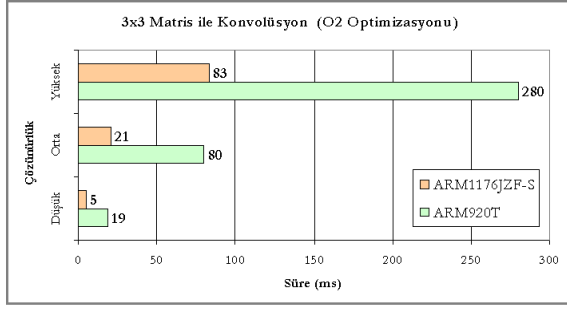
Günümüzde birçok firma, çeşitli çevre birimlerine sahip, ARM çekirdekli mikrodenetleyiciler üretmektedir. Çalışmamızın temel amacı ARM çekirdekli mikrodenetleyicilerin, işlem gücü açısından, TCAA'larda kullanımının uygunluğunu belirlemek olduğundan mikrodenetleyici yapısında kullanılan çevre birimlerin özel bir belirleyiciliği yoktur. Bu sebeple ARM9 ve ARM11 çekirdeğine sahip herhangi bir mikrodenetleyici bu testler için kullanılabilir.

ARM9 tabanlı mikrodenetleyicilerin performans ölçümünde FriendlyARM firmasının mini2440 geliştirme ortamı kullanılmıştır. Mini2440 geliştirme ortamında, ARM920T çekirdeğine sahip S3C2440A mikrodenetleyici ve 64MB SDRAM özelliğinde bellek bulunmaktadır. ARM11 tabanlı mikrodenetleyicilerin performans ölçümlerinde de yine FriendlyARM firması tarafından üretilen mini6410 geliştirme ortamı kullanılmıştır. Mini6410 geliştirme ortamında, ARM1176JZF-S çekirdeğine sahip S3C6410A mikrodenetleyici 128MB SDRAM bulunmaktadır.

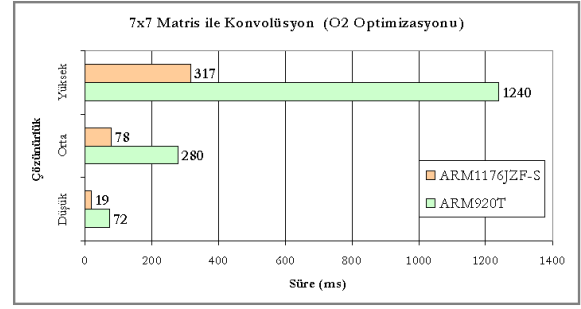
Mikrodenetleyicilerin performans ölçümleri, ilgili geliştirme ortamlarında herhangi bir işletim sistemi kullanılmadan doğrudan kod çalıştırılarak gerçekleştirilmiştir. Ayrıca testler sırasında mikrodenetleyiciler üzerinde bulunan ve kullanılmayan çevre birimleri kapatılarak, saf çekirdek performansı ölçülmüştür. Mikrodenetleyicilerin mimarilerine uygun en verimli kodun üretilebilmesi için ARM konsorsiyumu tarafından üretilen ARVDS 4.1 derleyici ortamı kullanılmıştır.

Mikrodenetleyiciler arasındaki performans farkını göstermek amacıyla yapılan konvolüsyon testlerinde iki temel kriterden faydalanılmıştır; *görüntü çözünürlüğü* ve *konvolüsyon matris boyutu*. TCAA'larda geçerli olan düşük çözünürlük için 160x120, orta çözünürlük için 320x240 ve yüksek çözünürlük için 640x480 seçilmiştir. Konvolüsyon matrisinin boyutu ise 3x3, 5x5 ve 7x7 olarak belirlenmiştir. Çözünürlükler ve matris boyutlarının tüm kombinasyonları için zaman açısından (-Otime) optimize edilmiş kodlar üretilerek test işlemi gerçekleştirilmiştir. Seçilen optimizasyon temel olarak döngü optimizasyonu, yaygın alt ifadelerin ayıklanması ve komutların düzenlenmesini içerir. Şekil 1, Şekil 2 ve Şekil 3'te, ARM920T ve ARM1176JZF-S çekirdekleri için yukarıda belirtilen çözünürlüklerdeki görüntüler üzerinde gerçekleştirilen konvolüsyon işlemlerinin süreleri karşılaştırmalı olarak verilmiştir.

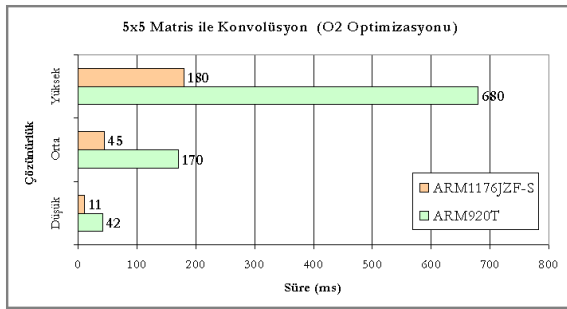
Her iki çekirdeğin, testler için belirlenen çalışma frekansı düşünüldüğünde, ARM11'in(533MHz) ARM9'a(400MHz)



Şekil 1: Düşük, orta ve yüksek çözünürlükteki görüntülerin üzerinde yapılan konvolüsyon işlemine ait süreler (3x3 Matris Boyutu)



Şekil 3: Düşük, orta ve yüksek çözünürlükteki görüntülerin üzerinde yapılan konvolüsyon işlemine ait süreler (7x7 Matris Boyutu)



Şekil 2: Düşük, orta ve yüksek çözünürlükteki görüntülerin üzerinde yapılan konvolüsyon işlemine ait süreler (5x5 Matris Boyutu)

göre 1.33 kat daha hızlı sonuç üretmesi beklenmektedir. Ancak test sonuçları ARM11'in yaklaşık dört kat daha hızlı çalıştığını göstermektedir. Aradaki ekstra hız farkını, bir önceki bölümde bahsedilen ARM1176JZF-S'nin özellikle komut seti, işhattı ve veriyolundaki mimari iyileştirmeler sağlamaktadır.

ARM1176JZF-S hız dışında enerji tüketimi açısından da ARM920T'ye göre çok daha avantajlıdır. ARM920T MHz başına 0.25mW harcarken ARM1176JZF-S 0.208mW tüketmektedir. Her iki işlemciye ait her bir test sırasında gerçekleşen ortalama enerji tüketim değerleri Tablo 1'de verilmiştir. Tablo 1 incelendiğinde ARM1176JZF-S mikrodnetleyicisinin aynı işlem için ARM920T'ye göre 3 ila 4 kat arasında daha az enerji harcadığı görülmektedir.

Tablo 1: ARM920T ve ARM1176JZF-S çekirdeklerinin enerji tüketimi açısından karşılaştırılması (mJ)

Çözünürlük	ARM920T ¹			ARM1176JZF-S ²		
	3x3	5x5	7x7	3x3	5x5	7x7
Düşük	1.9	8	28	0.55	2.33	9.20
Orta	4.2	17	68	1.22	4.99	19.96
Yüksek	7.2	28	124	2.11	8.65	35.14

¹ 400MHz, 0.25mW/MHz.

² 533MHz, 0.208mW/MHz.

5. SONUÇ

TCAA uygulamalarında baz istasyonları ile haberleşmeyi en düşük seviyede tutarak enerji tasarrufu sağlamak için ham verinin işlenerek anlamlı bilgiye dönüştürülmesi kaçınılmazdır. ARM7 tabanlı mikrodnetleyicilerin kullanıldığı düğümler ile bu hedef doğrultusunda çalışmalar yapılmışsa da istenilen performans seviyesine ulaşılamamıştır. Bu çalışmada, yeni nesil ARM9 ve ARM11 çekirdeklerine sahip mikrodnetleyicilerin temel görüntü işleme yetenekleri incelenmiştir. İncelemenin sonucunda her iki çekirdeğin performansının bahsi geçen uygulamalar için yeterli olduğu ancak ARM11 çekirdeğinin sahip olduğu mimari özellikler ile ARM9'a göre hem işlem gücü hem de enerji tüketimi açısından 3 ila 4 kat daha verimli olduğu görülmüştür.

6. KAYNAKÇA

- [1] I. T. Almalkawi, M. Guerrero Zapata, J. N. Al-Karakı, and J. Morillo-Pozo, "Wireless multimedia sensor networks: Current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662-6717, 2010.
- [2] I. F. Akyıldız, T. Melodia, and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks (Elsevier)*, vol. 51, no. 4, pp. 921-960, mar 2007.
- [3] R. Zilan, J. M. Barcelo-Ordinas, and B. Tavli, "Wireless systems and mobility in next generation internet." Berlin, Heidelberg: Springer-Verlag, 2008, ch. Image Recognition Traffic Patterns for Wireless Multimedia Sensor Networks, pp. 49-59.
- [4] D. McIntire, K. Ho, B. Yip, A. Singh, W. Wu, and W. J. Kaiser, "The low power energy aware processing (leap) embedded networked sensor system," in *Proceedings of the 5th international conference on Information processing in sensor networks*, ser. IPSN '06, 2006, pp. 449-457.
- [5] I. Akyıldız, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588-1605, Oct. 2008.
- [6] "ARM Website," <http://www.arm.com/>.
- [7] M. G. Benjamin and D. Kaeli, "Stream image processing on a dual-core embedded system," in *Proceedings of the 7th international conference on Embedded computer systems: architectures, modeling, and simulation*, ser. SAMOS'07, 2007, pp. 149-158.
- [8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.