



## OPENCL İLE GERÇEK ZAMANLI DOĞAL TAŞ TASNİFİ VE PERFORMANS ANALİZİ

Abdullah Serdar ÖNDER<sup>1\*</sup>, Tuna GÖKSU<sup>2</sup>

<sup>1</sup>Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Bölümü, Isparta, Türkiye

<sup>2</sup>Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Isparta, Türkiye

### Anahtar Kelimeler

OpenCL,  
Renk uzayları,  
Doğal taş tasnifi,  
Performans analizi.

### Öz

Doğal taş üretim tesislerinde, paketlenen her kasa, taşın aynı seleksiyonlarını içermesi gerekmektedir. Taşların doğru seleksiyona ait kasalara yerleştirilememesi hatalı sınıflandırma sonuçlarını doğurabilmektedir. Bu çalışma doğal taş tasnifini, personel yorumuna bırakmadan hızlı ve doğru şekilde sağlanması amacı ile gerçekleştirilmiştir. Elde edilen görüntüler 18 farklı renk uzayında gözlemlenmiş, sayısal olarak elde edilen veriler incelenmiş ve seleksiyon ayırımında kullanılacak renk uzayı dönüşümleri belirlenmiştir. OpenCL yazılım çatısı ile paralelleştirilmiş uygulama CPU, GPU, FPGA üzerinde çalıştırılmış ve bu platformlarda performans analizleri gerçekleştirilmiştir. GPU, CPU'dan 4.5 kat ve FPGA'den 26 kat daha hızlı işlem yapmıştır. Görüntüler aynı ortamda on kez tekrarlanarak elde edilmiş ve tasniflendirme işlemi %100 doğrulukla sonuçlandırılmıştır.

## REAL-TIME NATURAL STONE CLASSIFICATION WITH OPENCL AND PERFORMANCE ANALYSIS

### Keywords

OpenCL,  
Color spaces,  
Natural stone classification,  
Performance analysis.

### Abstract

In natural stone production facilities, every box should contain the stones' same selection. The placement of stones in wrong selection boxes causes the classification results to be inaccurate. This study has been carried out with the aim of providing the natural stone classification straight and quickly in a way that can be done alone without staff. The images were observed in 18 different color spaces, the numerical data were obtained from each type of marble and color space transformations that can be used for selection are determined. The application parallelized with OpenCL software renderer was runned on the CPU, GPU, FPGA and a performance analyzes were performed on this platforms. The GPU is 4.5 times faster than the CPU and 26 times faster than the FPGA. The images were obtained by repeating ten times in the same experimental environment and the classification process was concluded with 100% accuracy.

### Alıntı / Cite

Önder, S., A., Göksu, T. (2019). OpenCL ile Gerçek Zamanlı Doğal Taş Tasnifi ve Performans Analizi, Mühendislik Bilimleri ve Tasarım Dergisi, 7(3), 689-698.

### Yazar Kimliği / Author ID (ORCID Number)

A. S. Önder, 0000-0002-3968-0774  
T. Göksu, 0000-0002-9028-6422

### Makale Süreci / Article Process

Başvuru Tarihi / Submission Date	31.01.2019
Revizyon Tarihi / Revision Date	09.05.2019
Kabul Tarihi / Accepted Date	15.05.2019
Yayın Tarihi / Published Date	15.09.2019

### 1.Giriş

Doğal taşlar, doğal yapıda oldukları için kendi türleri arasında fiziksel ve görsel farklılıklar gösterebilir. Doğal taş üretim tesislerinde üretim hattının en son safhası tasniflendirme işlemidir. Tasniflendirme

işleminin hatalı yapılması doğal taş montaj aşamasında görsel uyumsuzluk yaratmakta ve üretici ile müşteri arasında anlaşmazlık doğurmaktadır. Üretilen her paket, her türün aynı seleksiyonunu içermesi gerekmektedir.

\* İlgili yazar / Corresponding author: asonder86@gmail.com, +90-555-015-7432

Bu çalışmada, seçilmiş olan mermer türlerine ait görüntüler, farklı renk uzaylarında incelenerek özellik çıkarımları gerçekleştirilmiştir. Özellik çıkarım yöntemi belirlendikten sonra OpenCL yazılım çatısı ile oluşturulan yazılım paralelleştirilmiştir. Paralel mimarilere sahip CPU, GPU ve FPGA üzerinde performans analizleri gerçekleştirilmiştir.

İlk bölümde görüntülerin elde edildiği ortam ve kullanılan materyallerden bahsedilmiştir. Ardından önerilen yöntemlerle ilgili olarak renk ve renk uzayları, özellik çıkarımı, Paralel programlama ve OpenCL ve FPGA hakkında bilgiler verilmiştir. Dördüncü bölümde, elde edilen sonuçlar verilmiş ve performans analizleri karşılaştırılmıştır. Son bölümde ise, elde edilen sonuçlar benzer çalışmalarla birlikte yorumlanmış ve ileriki çalışmalar için öneriler sunulmuştur.

## 2. Bilimsel Yazın Taraması

Geniş bir yelpazede uygulamalarla uyumlu çalışan OpenCL'in yaygın kullanım alanlarından biri, görüntü ve video işlemedir. Görüntü işleme alanındaki en önemli zorluklardan biri, doğruluk ve gerçek zamanlı çalışma sağlamak için yüksek işlem gereksinimidir. Resim çözünürlüklerinin ve veri boyutlarının artışı, gerçek zamanlı performans elde etmeyi zorlaştırmaktadır (Sümerkan, H., D., 2014).

FPGA, elektronik tasarım ve programlama alanında son zamanlarda önemli bir çalışma ortamı sunmuştur. Paralel programlama yöntemlerinin gelişmesi ve daha geniş belleklerin daha az yer kaplaması FPGA üzerinde bellek ve işlemci hızı gibi kısıtlamaları azaltmış ve dolayısıyla hızlı çalışabilen iş hattı yapısına sahip algoritma modelleme yöntemleri gerçekleşmiştir (Greisen, P. vd., 2011).

Çil, 2015 yılında yapmış oldukları çalışmada FPGA üzerinde temel görüntü işleme tekniklerinden olan kenar bulma algoritmalarını kullanmıştır. Bu yöntemler Sobel, Prewitt, Canny, Gauss, LoG algoritmalarıdır. Yaptığı çalışmanın trafik uygulamalarında kullanılabilmesini ve ileriye gönük olarak da oluşturulmak istenen görüntü işleme kütüphanesi için bir temel olabileceğini belirtmiştir.

Saraç, 2016 yılında yapmış olduğu çalışmada, FPGA üzerinde gerçek zamanlı nesne algılama uygulaması yapmıştır. Terasic DE2-115 FPGA seti ve TRDB-DSM kamera modülü kullanılmıştır. Nesne algılama sistemi olarak 3x3 pencere boyutlarına sahip medyan filtre optimize edilmiştir.

Altuncu vd., 2018 yılında yapmış oldukları çalışmada FPGA üzerinde gerçek zamanlı olarak temel görüntü işleme algoritmalarını uygulamıştır. USB kameradan alınan görüntüler üzerinde Verilog donanım tanımlama dili kullanarak griye dönüştürme,

konvolüsyon ile kenar belirleme, bulanıklık giderme, keskinleştirme gibi uygulamalar gerçekleştirilmiştir.

FPGA, CPU ve GPU üzerinde OpenCL paralel mimarisi ile elde edilen performans ve başarımlarının analizlerinin görüntü işleme uygulamalarında incelendiği bazı tez çalışmaları aşağıda sunulmuştur.

Ayat vd., 2015 yılında yapmış oldukları çalışmalarında Cyclone V FPGA kartında görüntü işleme deneyi olarak bir Sobel filtre algoritmasının uygulanması için, OpenCL programlama platformunu kullanmışlar; bu platformun taşınabilirliğini kullanarak, çekirdek kodunun performansını, farklı görüntü ve çekirdek boyutları için GPU ve çok çekirdekli CPU uygulamalarına göre karşılaştırmışlardır. Çıkan verilere göre her cihazın kendi optimizasyonunu gerektirdiği ve aynı kodun bu optimizasyonlar olmadan çok iyi performans göstermeyeceğini belirtmişlerdir.

Pereira vd., 2016 yılında yapmış oldukları çalışmalarında CPU ve GPU platformları için Walsh-Hadamard dönüşümünün paralel uygulaması üzerine yapılan geliştirmede; OpenCL'in kod düzeyinde taşınabilir olduğu ancak CPU ve GPU için aynı kod kullanıldığı zaman performansın düşeceğini belirtmişler ve GPU üzerinde çalışan OpenCL kodun, çok çekirdekli bir CPU'da çalışan OpenCL koddan hızlı çalıştığını gözlemlemişlerdir.

Doğal taşların öznelik çıkarımı ve sınıflandırılmalarıyla ve ilgili yapılan tez çalışmaları aşağıda sunulmuştur.

Abadi ve Banihashemi 2015 yılında yapmış oldukları çalışmalarında dört gruptan 124 adet resim içeren bir dizi traverten taşının sınıflandırılmasını incelemişlerdir. Öznelik çıkarma yaklaşımı olarak toplam ve fark histogramı (Sum Difference Histogram, SDH) kullanılmış ve elde edilen özneliklere çeşitli sınıflandırma algoritmaları uygulanmıştır. Yapılan deney sonuçları önerilen yaklaşımın traverten taşlarında kabul edilebilir bir sınıflandırma performansına sahip olduğunu belirtmişlerdir.

Bianconi vd., 2015 yılında yapmış oldukları çalışmalarında granit yüzeylerini sınıflandırmada farklı renk uzayları kullanılmasının sınıflandırma üzerindeki etkisini ölçmüşlerdir. 25 farklı granit sınıfının karşılaştırıldığı çalışmada k-NN sınıflayıcısı kullanılarak farklı renk uzaylarından elde edilen özneliklerin performanslarında önemli bir farklılık gözlemlenmemiş, doğrusal sınıflayıcının kullanıldığı çalışmalarda Lab ve Luv uzaylarından elde edilen öznelikler diğerlerine oranla daha iyi sonuçlar verdiğini gözlemlemişlerdir.

Huang vd., 2016 yılında yapmış oldukları çalışmalarında 20 farklı granit görüntüsünü, gri seviye eş oluşum matrisleri, gabor filtreleri ve renk bilgisi

olarak HSV renk uzayından elde ettikleri öznitelikler ile birleştirerek destek vektör makineleri yardımı ile sınıflandırma işlemi gerçekleştirmişlerdir. Çalışmalarında renk bilgisi, HSV renk uzayında ki her bir renk kanalının histogram bilgileri 16, 4, 4 olacak şekilde kuantalanarak birleştirip tek bir vektör haline getirmişlerdir.

Literatür ve yapılan çalışmalar incelendiğinde, benzer projelerde başka platformlarda renk uzay dönüşümleri gözlemlenmektedir. Yine görüntü işleme teknikleri aynı platformda farklı konular üzerinde geliştirilmektedir. Doğal taşların renk uzay dönüşümü kullanılarak OpenCL mimarisi ile FPGA üzerinde tasarlanması ve sınıflandırılması bu alanda uygulanabilirliğe yönelik bu çalışmayı özgün kılmaktadır.

### 3. Materyal ve Yöntem

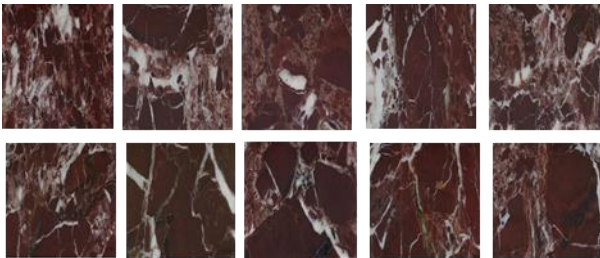
#### 3.1. Materyal

##### 3.1.1. İncelenen Doğal Taşlar

Çalışmada, doğal taş türü olarak Elazığ Vişne, Tundra grubu, Aksu grubu ve Manyas mermerleri kullanılmıştır. Dört farklı mermer için dokuz farklı seleksiyon için beşer adet örnek incelenmiştir. İncelenen örnek görüntüleri Şekil 1,2,3 ve 4'te sunulmuştur.



Şekil 1. Tundra Grubu Mermer Görüntüleri



Şekil 2. Elazığ Vişne Mermer Görüntüleri



Şekil 3. Manyas Mermeri Görüntüleri



Şekil 4. Aksu Grubu Mermeri Görüntüleri

##### 3.1.2. Deney Ortamı

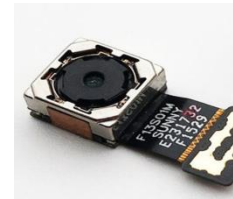
Görüntüler, dışarıdan ışık geçirmeyecek bir kapalı kutu içerisinde 3W gücünde 175 lümen ışık akısına sahip iki adet LED aydınlatma ile edilen suni aydınlatma ortamında sağlanmıştır. Görüntülerin elde edildiği ortam Şekil 5.'de verilmiştir.



Şekil 5. Görüntülerin Elde Edildiği Ortam

##### 3.1.3. Kamera

Görüntülerin elde edilmesi için, 3264x2480 piksel (8MP) çözünürlük, f/2.2 lens diyafram açıklığına sahip ve otomatik odaklanma yapabilen Vibe P1M P1A42 model kamera kullanılmıştır. Şekil 6.'da kullanılan kamera gösterilmektedir.



Şekil 6. Görüntülerin Elde Edildiği Kamera

### 3.1.4. Diğer Materyaller

Görüntüleri incelemede kullanılan CPU ve GPU'lar Tablo 1.'de teknik özellikleri ile birlikte sunulmuştur. Çalışmada incelenen FPGA'in özellikleri şöyledir:

- FPGA Geliştirme Kartı: Altera DE1-SoC, 800 MHz çift çekirdekli ARM Cortex-A9 işlemci, 1GB DDR3 SDRAM bellek, 4450 Kbit gömülü bellek, 6 adet kademeli PLL, 2 tane sabit bellek denetleyici.

### 3.2. Önerilen Yöntem

#### 3.2.1. Öznitelik Çıkarımı

Görüntüler üzerinde özellik çıkarımı, görüntüler üzerinde sayısal işlemler diğer adıyla görüntü işleme ile gerçekleştirilebilir. Görüntü üzerinde işlemler yapılması, sayısal değerleri gerektirmektedir.

##### 3.2.1.1. Renk Analizine Göre Öznitelik Çıkarımı

Piksel yoğunluğunun istatistiksel özelliklerinden yararlanılarak oluşturulan bir özellik çıkarım yöntemidir (Kemaloglu, N. vd., 2018). En çok kullanılan morfolojik işlemler aşağıda sıralanmıştır.

$$Enerji = \sum_{ij} P_{ij}^2 \quad (1)$$

$$Kontrast = \sum_{ij} |i - j|^2 P_{ij} \quad (2)$$

$$Korelasyon = \sum_{ij} (i, j) P_{ij} \mu_x \mu_y / \sigma_x \sigma_y \quad (3)$$

$$Homojenlik = \sum_{ij} P_{ij} / (1 + |i - j|) \quad (4)$$

$$Entropi = \sum_{ij} P_{ij} \log_{ij} \quad (5)$$

##### 3.2.1.2. Doku Analizine Göre Öznitelik Çıkarımı

Özellik çıkarımı, bir renk uzayında yapılan renk analizleri ile sağlanamadığı durumlarda doku analizi yapılarak görüntü farklı renk uzaylarına dönüştürülür. Diğer renk uzaylarında da renk analizine göre özellik çıkarımı yapılabilmektedir. (Kemaloglu, N. vd., 2018).

**Tablo 1.** İncelenen CPU ve GPU'ların Teknik Özellikleri

Teknik Ölçütler	CPU Teknik Özellikleri			Teknik Ölçütler	GPU Teknik Özellikleri				
	Intel i5 7200U	Intel i5 8250U CPU	Intel i7 8550U		Intel® HD Graphics 620	Intel® UHD Graphics 620	NVIDIA GeForce 940MX	NVIDIA GeForce GTX 570M	NVIDIA Quadro K2200
Çekirdek Sayısı	2	4	4	Çekirdek hızı	300-1050 MHz	300-1150 MHz	1120 MHz	1070 MHz	1124 MHz
İş Parçacığı Sayısı	4	8	8	Bellek transfer hızı	1067 MHz	1067 MHz	1001 MHz	750 MHz	1253 MHz
Bellek Transfer Hızı	2133 MHz	2168 MHz	2426 MHz	Verimli bellek transfer hızı	-	-	2002 MHz	3800 MHz	4000 MHz
Veri yolu Genişliği	64 bit	64 bit	64 bit	Veri yolu Genişliği	128 bit	128 bit	64 bit	64 bit	128 bit
Çalışma Frekansı (min)	2,5 GHz	1,6 GHz	1,8 GHz	Bant genişliği	-	-	16 GB/s	72 GB/s	80,19 GB/s
Çalışma Frekansı (max)	3,1 GHz	3,4 GHz	4 GHz	Doldurma Hızı (Texture Fill Rate)	25,20 GTexel/s	25,20 GTexel/s	29,81 GTexel/s	32,20 GTexel/s	44,96 GTexel/s

Bu çalışmada, görüntüler 18 farklı renk uzayında incelenmiş. Tablo 2.'de incelenen renk uzayları verilmiştir.

Mermer türleri için renk uzaylarından elde edilen veriler incelendiğinde, CMYK renk uzayının mermer türlerinin kendi içinde seleksiyonlarına ayrılabilmesi için uygun ve yeterli olduğu tespit edilmiştir.

CMYK renk uzayı cam göbeği (cyan), eflatun (magenta), sarı (yellow) ve siyah (key) renklerini ifade etmektedir. CMY renk uzayı cam göbeği, eflatun ve sarı renklerin belirli oranlarda birleşmesi ile oluşmaktadır. Bu üç renk RGB renklerinin tamamlayıcı olarak ifade edilir (Tosun, O., 2015).

$$[C; M; Y] = L - [R; G; B] \quad (6)$$

**Tablo 2.** İncelenen Renk Uzayları

İncelenen Renk Uzayları		
RGB	YPbPr	LUV
YIQ	YUV	LCH
HSV(HSB)	HSL	Cat02 LMS
YCbCr	HSI	CMYK
JPEG-YCbCr	XYZ	NTSC
YDbDr	Lab	OHTA

(6) eşitliğinde görüldüğü gibi RGB uzayından CMY renk uzayına geçmek istersek beyaz renkten bütünleyeni çıkarmak gerekmektedir (Tosun, O., 2015). L, 255 sayısal değerini yani beyazı ifade etmektedir.

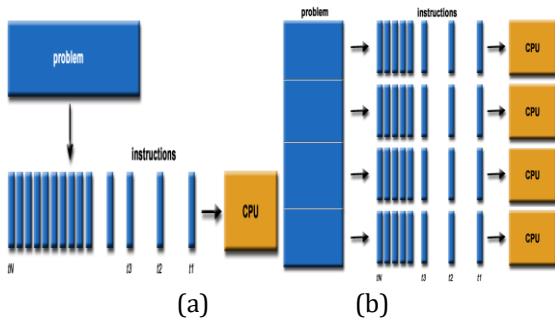
CMYK renk uzayı, boya tabanlı bir renk uzayıdır. Yazıcı ve matbaalar için geliştirilmiştir. Üç renk (cam göbeği, eflatun ve sarı) siyah rengi oluşturması gerekirken renk şiddetleri yetersizliğinden kaynaklı maliyeti düşürmek için siyah renk sonradan ilave edilmiş ve CMYK renk uzayı ortaya çıkmıştır (Tosun, O., 2015).

$$K = \min(C, M, Y) \quad (7)$$

(7) eşitliğinde gösterildiği gibi CMYK renk uzayında K (siyah) değeri C, M ve Y sayısal değerlerinin minimum değeri ile elde edilmektedir (Tosun, O., 2015).

### 3.2.2. Paralel Programlama

Paralel programla, bir problemin çözümünde birden çok hesaplama kaynağının aynı anda kullanılmasıdır. Seri programlamada birim zamanda bir işlem için bir işlemci çalışırken, paralel programlamada birden fazla işlem için birden fazla işlemci çalışmaktadır (Herlihy, M. vd., 2012).



**Şekil 7.** (a) Seri programlama, (b) Paralel programlama

Şekil 7.'de görüldüğü gibi aynı zaman diliminde, seri programlama bir işlemci ile bir işlem yaparken, paralel programlamada dört işlemci ile dört işlem yapılmaktadır. Paralel programlamanın en büyük avantajı, uygulama çözüm süresini minimum süreye indirmesidir. Çünkü paralel programlama, uygulamayı eş zamanlı parçalara böler ve her bir parçanın farklı

işlemci üzerinde işlenmesini sağlar (Gaster, B. vd., 2013).

### 3.2.3. OpenCL

OpenCL, C++ tabanlı heterojen platform üzerinde istenilen cihazlar üzerinde kodlar oluşturulabilen bir açık kaynak kodlu yazılım çatısıdır (Gaster vd., 2013). OpenCL, C++ programlama dili ile programın geliştirilebilmesine ve OpenCL tarafından sağlanan paralelleştirme gibi ek özelliklerin kullanılmasına olanak tanır. Çoğu uygulama alanında uyumlu çalışan OpenCL'in en yaygın olarak kullanıldığı alanlardan biri görüntü ve video işlemedir. OpenCL yazılım platformu, birçok donanım platformu tarafından desteklenmektedir (Eskikaya, B., 2012).

OpenCL cihazı CPU, GPU, DSP ve FPGA gibi donanımlardan biridir. Program nesnelere, programın kodunu ve derlenmiş halini, derlenme kaydı ile program aygıt tipi gibi bilgilerini içeren OpenCL programını temsil eden veri şeklidir. Çekirdek fonksiyonları, OpenCL destekli aygıtların hesaplama ünitelerinde çalıştırılması için yazılmış fonksiyonlardır. Derlendiğinde bir çekirdek oluşur (Eskikaya, B., 2012).

#### 3.2.3.1. OpenCL Çekirdek Kodları

Bu bölümde görüntülerin RGB renk uzayından CMYK(K) renk uzayına dönüştürülmesi için OpenCL ile paralelleştirilmiş çekirdek fonksiyonu gösterilmektedir.

**Tablo 3.** OpenCL Çekirdek (Kernel) Kodları

```

_kernel void rgb2k( _global uchar * input,
                  _global uchar * output )
{
  int w = get_global_size(0);
  int h = get_global_size(1);
  int y = get_global_id(1);
  int x = get_global_id(0);
  input[(y*w+x)*3] = max( input[(y*w+x)*3], input[(y*w+x)*3+1] );
  input[(y*w+x)*3] = max( input[(y*w+x)*3], input[(y*w+x)*3+2] );
  output[(y*w+x)] = 255 - input[(y*w+x)*3];
}

```

### 3.2.4. FPGA

FPGA, üretiminden sonra istenen fonksiyona göre donanımsal olarak tasarlanabilen tümleşik devre olarak ifade edilebilir. FPGA, kısaltma ifadesini "Field Programmable Gate Array" kelime grubunun baş harflerinden almaktadır. Bu ifade, sahada programlanabilir yani mantıksal fonksiyonları tasarımcı tarafından sahada oluşturulabilen kapı dizileri anlamında gelmektedir (Brown, S. vd., 1992). FPGA, tasarlanabilmesi için mantık kapıları birbirinden bağımsız, bağlantıları tasarımcıya

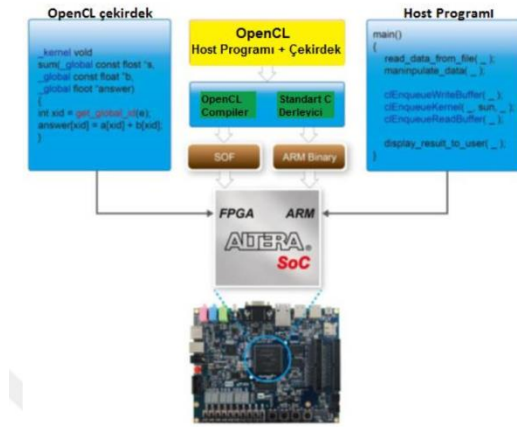
bırakılan serbest işlenmemiş tümleşik devre olarak üretilmektedir.

FPGA paralel olan doğal yapısı ile OpenCL'in paralel işlem yapma yetisiyle birlikte sorunsuz ve uyumlu olarak çalışabilmektedir.

### 3.2.5. OpenCL ve FPGA

Çekirdek (Kernel) olarak adlandırılan paralelleştirilmek istenen iş fonksiyonunun, FPGA üzerinde yazılımcı tarafından yazılımcıya düşük seviyeli HDL dilleri öğrenme zorunluluğuna bırakmadan OpenCL ile C dili ana yapısı kullanılarak kolaylıkla programlanabilmesini sağlamaktadır.

Çoğu FPGA üreticisi, FPGA'lar üzerinde OpenCL ile geliştirme için yazılım geliştirme araçları sağlar (Moore, A., 2014). Bunlardan biri de Bu çalışmada kullanılacak olan Altera entegre içinde sistem (SoC) FPGA ile oluşturulmuş sağlam bir donanım tasarım platformu olan DE1-SoC modelidir. DE1-SoC'de geliştirilecek bir OpenCL yazılım projesi OpenCL çekirdeği (kernel) ve ev sahibi program (host program) dan oluşur. Çekirdek, SoC FPGA'in FPGA kısmı üzerinde gerçekleştirilir ve ana program, SoC FPGA'in CPU kısmında bulunur. Ana programı, Windows veya Linux'a kurulu Altera SoC EDS tarafından çapraz derleyici ile derlenir ve çekirdek, Windows veya Linux'a kurulmuş olan Quartus ve OpenCL SDK tarafından geliştirilir. Şekil 8.'de Altera Soc OpenCL mimarisi gösterilmiştir.



Şekil 8. Altera Soc OpenCL mimarisi (Faisal, M., R., 2018)

## 4. Araştırma Bulguları

### 4.1. Deneysel Sonuçlar

Elde edilen görüntüler RGB renk uzayından CMYK renk uzayına dönüştürüldükten sonra elde edilen veriler Tablo 4.'de verilmiştir. Çıkan değerlere göre her mermer grubu için eşik değeri belirlenmiş ve sınıflandırma işlemi gerçekleştirilmiştir. Görüntüler

aynı ortamda on kez tekrarlanarak elde edilmiş ve tasniflendirme işlemi %100 doğrulukla sonuçlandırılmıştır.

Tablo 4. CMYK(K) Renk Uzay Dönüşümünde Elde Edilen Veriler

Mermer Türleri	Cmyk(K) Renk Uzay Dönüşümünde Elde Edilen Veriler	
	Ortalama değer / Standart Sapma	Eşik Değeri
Elazığ Vişne Siyahlı	177,24	Eşik değeri olarak 170 belirlenmiştir.
Elazığ Vişne Temiz	163,67	
Manyas Siyahlı	7,262 (s.s)	Eşik değeri olarak 5 belirlenmiştir.
Manyas Temiz	2,388 (s.s)	
Tundra Gray	72,012	Eşik değeri olarak 90 belirlenmiştir.
Tundra Blue	106,26	
Nordic Gray	65,83	Eşik değeri < 75 95 > Eşik değeri > 75 olarak belirlenmiştir.
Ice Gray	85,74	
Earth Gray	110,51	
Tüm mermer seleksiyonları için elde edilen sonuçların ortalaması alınmıştır.		

\* s.s = standart sapma

Intel i5 7200U CPU üzerinde seri olarak gerçekleştirilen sınıflandırma işlemi, OpenCL mimarisi ile paralelleştirilerek aynı CPU üzerinde çalıştırılmıştır. OpenCL ile paralelleştirilen çekirdek fonksiyonu çalışma süresi ile, seri programda çekirdek fonksiyonuna eş değer olan kısmın çalışma süreleri kıyaslanmış ve Tablo 5.'te verilmiştir.

Tablo 5. Seri ve Paralel İşlem Süreleri

512x512 Çözünürlük	RGB → CMYK (K) Dönüşüm İşlem Süreleri (ms)	
	Intel i5 7200U CPU üzerinde seri işlem süresi	Intel i5 7200U CPU üzerinde paralel işlem süresi
Elazığ Vişne	1,058	0,421
Tundra grubu	1,056	0,397
Manyas	1,099	0,404
Aksu grubu	1,059	0,42
Ortalama İşlem Süresi	1,068	0,41

Paralel programlama ile yaklaşık 2,6 kat performans kazanımı gözlemlenmiştir. Bu farkın en temel nedeni ise seri işlemden tek çekirdek üzerinde program çalıştırılırken paralel işlemden 2 çekirdek (4 iş parçacığı) üzerinde işlem dağıtılarak çalıştırılmaktadır. Bu beklenen bir performans kazanımıdır.

Tablo 1.'de özellikleri verilen farklı işlemciler üzerinde çekirdek fonksiyonu çalıştırılmış paralel işlem sürelerinin kıyaslanması amacı ile performans analizi gerçekleştirilmiştir. Tablo 6.'da elde edilen işlem süreleri verilmiştir.

**Tablo 6.** CPU'lar Üzerinde Paralel İşlem Süreleri

512x512 Çözünürlük	RGB → CMYK (K) Dönüşüm İşlem Süreleri (ms)		
	Intel i5 7200U CPU üzerinde paralel işlem	Intel i5 8250U CPU üzerinde paralel işlem	Intel i7 8550U CPU üzerinde paralel işlem
Elazığ Vişne	0,421	0,324	0,294
Tundra grubu	0,397	0,327	0,295
Manyas	0,404	0,324	0,308
Aksu grubu	0,42	0,33	0,289
Ortalama İşlem Süresi	0,41	0,326	0,297

Intel i5 8250U ve i7 8550U işlemcileri üzerinde elde edilen sonuçlar bu işlemcilerin i5 7200U işlemcisine kıyasla daha yüksek performans sergilediğini göstermektedir. Bunun en temel nedeni işlemci çekirdek sayısının paralel işlem performansını doğrudan etkilemesidir. Intel i5 7200U modeli fiziksel çift çekirdeğe sahipken diğer iki işlemci fiziksel dört çekirdeğe sahiptir. Intel i5 8250U işlemcisi i5 7200U işlemcisine kıyasla yaklaşık %21 lik bir performans kazanımı gösterirken, i7 8550U işlemcisi ise yaklaşık %28 lik bir kazanım elde etmiştir. Çekirdek sayısı iki kat artmasına rağmen performansta iki kat artışın gözlenmemesine sebep olarak, çekirdeklerin ana belleğe erişmek için kullandıkları veri yolunun çekirdekler arasında ortak kullanılıyor olması gösterilebilir. Shen vd., 2013 yılında yaptıkları çalışmalarında farklı çekirdek sayılarına sahip CPU'ların performans analizlerini gerçekleştirmişlerdir. 8 çekirdekli 2.40 GHz Intel Xeon E5620 model CPU ile 12 çekirdekli 2.67 GHz Intel Xeon X5650 model CPU üzerinde farklı çalışmalar gerçekleştirmiş ve performanslarını karşılaştırmışlardır. Farklı metotlara göre farklı sonuçlar elde etmişlerdir. Çekirdek sayısındaki 1,5 kat artışa karşılık performansta %7 ile %17 arasında kazanım elde etmişlerdir.

Intel i5 8250U ile i7 8550U aynı çekirdek sayısına sahip olmasına rağmen farklı çekirdek hızlarında işlem yapabilmektedir. Intel i7 8550U işlemcisi i5 8250U modeline göre daha yüksek işlemci ve bellek hızına sahiptir, aynı zamanda üretim teknolojileride farklılık göstermektedir. Tablo 7.'de bu frekans değerinde ki farklılığın işlemci performansını pozitif yönde etkilediği (yaklaşık % 9 performans kazanımı) görülmektedir.

Tablo 7.'de ise farklı GPU lar üzerinde çekirdek fonksiyonu çalıştırılmış ve elde edilen paralel işlem süreleri verilmiştir. Intel GPU'lar arasında birbirine çok yakın bir performans sonucu elde edilmiştir. Bunun nedeni aynı GPU için UHD modelinin azami çalışma frekansı HD modelinden 100 MHz daha yüksek olmasıdır. Intel GPU performansı harici ekran kartları ile kıyaslandığında GT 940MX modeline göre performansının daha hızlı olduğu görülürken GTX

570M ve QUADRO K2200 modeline göre ise performansının daha yavaş olduğu görülmektedir.

Intel GPU, NVIDIA GT940 MX modeline göre daha yüksek bellek transfer hızına ve daha yeni bir teknolojiye sahiptir. GTX 570M ve KK2200 modellerine göre ise çok yakın değerlerde işlemci hızına sahip olmasına rağmen çok ciddiye alınabilecek bir bellek transfer hızı farkı ortaya çıkmaktadır. Bellek transfer hızı ile pozitif yönde GTX 570M ve KK2200 modeli GPU'lar üzerinde en yüksek performans sonuçları elde edilmiştir.

**Tablo 7.** Farklı GPU'lar Üzerinde Paralel İşlem Süreleri

512x512 Çözünürlük	RGB → CMYK (K) Dönüşüm İşlem Süreleri (ms)				
	Intel HD Graphics 620	Intel UHD Graphics 620	NVIDIA GT940 MX	NVIDIA GTX 570M	NVIDIA QUADRO K2200
Elazığ Vişne	0,089	0,088	0,15	0,036	0,032
Tundra grubu	0,085	0,082	0,13	0,039	0,03
Manyas	0,09	0,09	0,15	0,034	0,03
Aksu grubu	0,091	0,09	0,16	0,035	0,032
Ortalama İşlem Süresi	0,089	0,087	0,14	0,036	0,031

Bu çalışmada elde edilen işlem süreleri tüm program içerisinde paralel olarak işlenen bölümü içermektedir. Bu bölüm OpenCL ile çalıştırılan çekirdek fonksiyon süresidir. Bu çekirdek fonksiyonu görüntünün paralel olarak işlenmesini kapsamaktadır.

İncelenen 512x512 çözünürlükteki dört mermer türü için elde edilen çekirdek fonksiyon sürelerinin ortalaması alınarak bu çalışmada incelenen tüm paralel platformlarda elde edilen performans analizi histogram grafiği olarak Şekil 9.'da sunulmuştur.

Bu çalışmada incelenen paralel platformlarda elde edilen ortalama performanslara göre GPU'ların CPU'lara göre yaklaşık 4.5 kat, FPGA'ye göre yaklaşık 26 kat daha performanslı olduğu görülmektedir.

## 5. Sonuç ve Tartışma

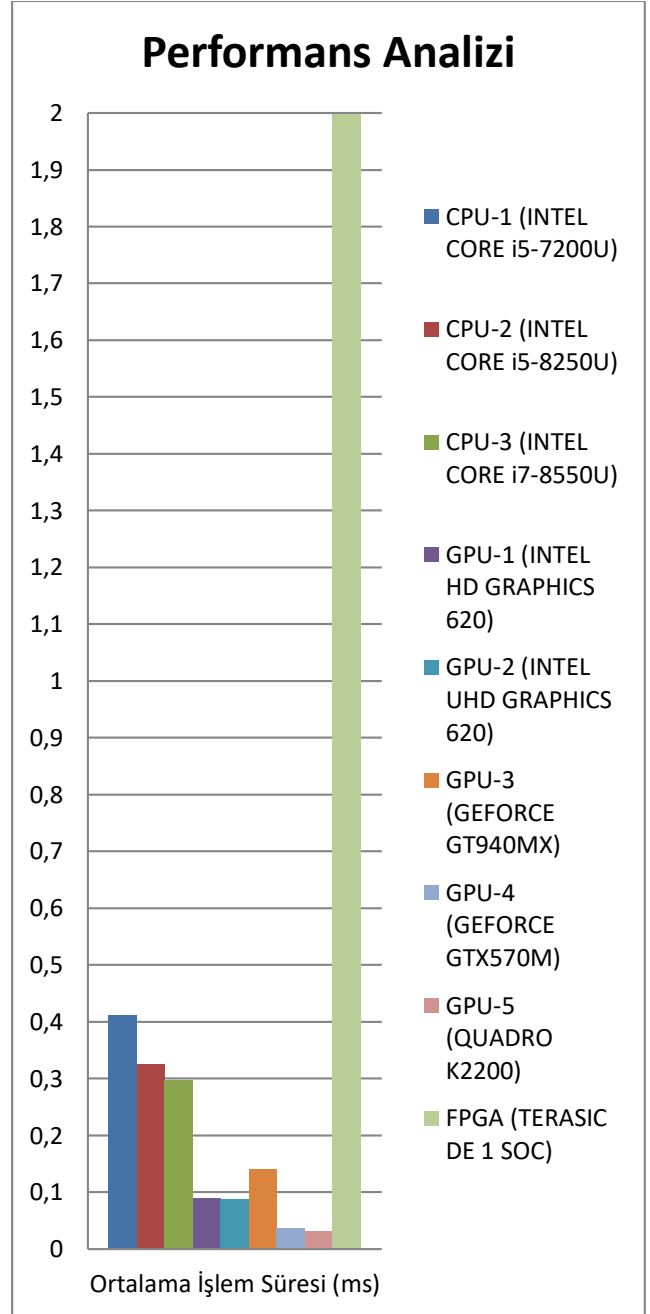
Bu çalışmada geliştirilen OpenCL uygulaması için, GPU'ların hem CPU hem de FPGA'den daha performanslı çalıştığı gözlenmiştir. Genel itibariyle çalışmada kullanılan CPU ve GPU'ların çalışma frekansı SoC FPGA geliştirme kartı işlemcisinin çalışma frekansından daha yüksektir. Aynı zamanda SoC FPGA geliştirme kartı için işlemci ile FPGA arasında veri aktarımını sağlayan veri yollarının

hızları ve genişlikleri, bilgisayarlarda CPU ile GPU arasındaki veri aktarımını sağlayan veri yolları ile kıyaslandığında daha az gelişmiştir. FPGA, GPU ile işlevleri noktasında kıyaslandığında daha az özelleşmiş bir cihazdır. GPU, esas itibari ile görüntüler ile işlem yapmak üzere mimarisi özelleştirilmiş olması sebebi ile bu çalışmadaki uygulamada en yüksek başarıyı vermiştir. GPU'ların sahip olduğu paralel işleme mimarisi, CPU'lara oranla dikkate değer performans farklılıkları ortaya çıkarmıştır. CPU'lar kendi içlerinde kıyaslandıklarında fiziksel çekirdek sayısının ve çalışma frekansının paralel çalışabilecek uygulamalar için dikkate alınması gereken özellikler olduğu, GPU'lar için bellek transfer hızının performansa tesiri olduğu gözlenmiştir.

Bu çalışmadaki analizler, işlemci frekanslarının en yüksek değerlerinde gerçekleştirilmiştir. CPU, GPU ve FPGA mimarilerinin birbirlerine göre kıyasla sahip oldukları pozitif ve negatif özellikler vardır. Bu da bu özelliklerin optimizasyonunun yapılması gerekliliğini ortaya çıkarmaktadır. Herhangi bir mimari için sadece işlemci hızına veya sadece bellek transfer hızına bakılarak bir değerlendirme yapmak doğru olmayacaktır. Her mimari en iyi optimizasyona sahipken en yüksek performansı sağlayacaktır. Bu sonuç ile, veri miktarı fazlalığına göre karmaşık yapıları uygulamalar için, en iyi performansı belirli bir paralel yapının veremeyeceği sonucunu da çıkartılabilir.

Chen vd. (2013), yapmış oldukları çalışmalarında elde ettikleri seri ve paralel işlem performans analiz sonucunun benzeri bu çalışmada da gözlemlenmiş bir sonuç olarak karşımıza çıkmıştır. Asano vd. (2009) ve Wu vd. (2014), yapmış oldukları çalışmalarında sade hesaplama yöntemi olan veya az veri işlemi gerektiren çalışmalarda GPU performansının CPU ve FPGA'ye göre daha yüksek olduğunu gözlemlemiştir. Bu çalışmada, elde edilen performans analizleri renk uzay dönüşümü işlemi ve işlenen veri miktarı olarak incelendiğinde GPU'nun CPU ve FPGA'den daha performanslı olması bu çalışmada beklenen bir sonuçtur.

Bu çalışma, daha çok renk ayrıtısına sahip ve ayırt edilmesi çok zor doğal taşlar için de geliştirilebilir. Bu çalışma, farklı renk uzaylarını birleştirebilen ve insan gözünün ayırt etmekte zorlanabileceği seleksiyon farklılıklarını ayırt edebilen bir yazılım oluşturmak için genişletilebilir. Genişletilen bu yazılım ile işlem süreleri ve doğruluk analizleri yapılabilir. Elde edilen işlem süresi dikkate alınarak farklı platformlar bu işletmelerde maliyet analizi yapılarak kullanılabilir. Bu çalışma, seleksiyon sınıflandırması dışında, tekstil gibi tek tip üretim yapabilen fabrikaların üretim bölümünde hata tespitinin daha hızlı bir şekilde gerçekleştirilebilmesi amacıyla paralel olarak farklı platformlarda tasarlanmak istenen projelere öncülük edebilir.



**Şekil 9.** Paralel Platformlarda Performans Analizi

#### Conflict of Interest / Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

No conflict of interest was declared by the authors.

#### Kaynaklar

Abadi, M., S., E., Banihashemi, N. 2015. Automatic classification of travertine stones based on sum and difference histograms algorithm. 2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP).



- Altuncu, M., A., Kösten, M., M., Çavuşlu, M., A. & Şahin, S. 2018. "Temel Görüntü İşleme Uygulamalarının Düşük Maliyetli IP Çekirdek olarak FPGA Tabanlı Gerçeklenmesi", IEEE Xplore, SİU 2018 IEEE 26. Sinyal İşleme ve İletişim Uygulamaları Kurultayı – İzmir.
- Asano, S., Maruyama, T. & Yamaguchi, Y. 2009. Performance comparison of FPGA, GPU and CPU in image processing. 19th International Conference on Field Programmable Logic and Applications. 126-131.
- Ayat, S., O., Khalil-Hani, M., & Bakhteri, R. 2015. OpenCL Based Hardware-Software Co-Design Methodology for Image Processing Implementation on Heterogeneous FPGA Platform. IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 36-41.
- Banger, R., Kumar, B., 2013. OpenCL Programming by Example. Packt Publishing, 287s, Birmingham.
- Bianconi, F., Bello, R., González, E. & Fernández, A. 2015. On Comparing Colour Spaces From A Performance Perspective: Application to Automated Classification of Polished Natural Stones, in New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops, V. Murino, E. Puppo, D. Sona, M. Cristani and C. Sansone eds., Vol. 9281 of Lecture Notes in Computer Science, 71-78.
- Brown, S., Rose, J., 1992. Field Programmable Gate Array. Kluwer Academic Publishers, 287s, USA.
- Chen, D. & Singh, D. 2013. Fractal video compression in OpenCL: An Evaluation of CPUs, GPUs, and FPGAs as Acceleration Platforms. 18th Asia and South Pacific Design Automation Conference (ASP-DAC). 297-304.
- Çelik, A., R., 2013. Görüntü İşleme Algoritmalarının FPGA Donanımı Üzerinde Gerçeklenmesi. Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 53s, Kahramanmaraş.
- Çil, M., M., 2015. Temel Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi İstanbul Teknik Üniversitesi, Fen bilimleri Enstitüsü, YL Tezi, 65s, İstanbul.
- Eskikaya, B., 2012. Distributed OpenCL – OpenCL Platformunun Ağ Ölçeğinde Dağıtılması. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 49s, İstanbul.
- Faisal, M., R., 2018. OpenCL ile FPGA Üzerinde Güvenlik Duvarının Gerçeklenmesi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 68s, Isparta.
- Gaster, B., Howes, L., Kaeli D., R., Mistry, P. & Schaa, D., 2013. Heterogeneous Computing with OpenCL: Revised OpenCL 1.2 Ed. Morgan Kaufmann Publishing, 291s, USA.
- Greisen, P., Heinzle, S., Gross, M., & Burg, A., P., 2011. An FPGA-Based Processing Pipeline For High-Definition Stereo Video. Erişim Tarihi: 24.08.2018. <https://jivp-eurasipjournals.springeropen.com>
- Gülmez, S., 2016. Asenkron Motorlarda Fpga ile Sensörsüz Gerçek Zamanlı Hız Tespiti, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 68s, Isparta.
- Herlihy, M., N., Shavit, 2012. The Art of Multiprocessor Programming, Revised Reprint, Elsevier, 536s, USA.
- Huang, S., Zhong, S. & Chen, K. 2016. A Novel Method of Stone Surface Texture Image Recognition. IEEE International Conference on Signal and Image Processing (ICSIP). 146-150.
- Kemaloglu, N., Aydogan, T., & Metlek, S., 2018. Classification of Travertine Tiles with Supervised and Unsupervised Classifiers and Quality Control, International Journal of Engineering and Technology, 10(3), 221-223.
- Moore, A., 2014. FPGAs for Dummies-Fpgas Keep You Moving in a Fastchanging World, Wiley John & Sons Publishing, 44s, USA.
- Pereira, Pedro, M., M., Domingues, P., Rodrigues, Nuno, M., M., Falcao, G. & Faria, S., M., M., 2016. Optimized Fast Walsh-Hadamard Transform on OpenCL-GPU and OpenCL-CPU, Proc IPTA 2016 - 6th International Conference on Image Processing Theory, Tools and Applications IPTA'2016, Oulu, Finland, Vol. 1, 1-8.
- Saraç, M., R., 2016. FPGA Kullanarak Gerçek Zamanlı Görsel Nesne Algılama, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, Elazığ.
- Shen, J., Fang, J., Sips, H., Varbanescu, A., L., 2013. An application-centric evaluation of OpenCL on multi-core CPUs, Parallel Computing, 39, 834-850.
- Sümerkan, H., D., 2014. Gömülü Sistemler Üzerinde OpenCL Tabanlı Görüntü İşleme Kütüphanesi ve Kernel Füzyon. TOBB Ekonomi ve Teknoloji Üniversitesi, Fen bilimleri Enstitüsü, YL Tezi, 69s, Ankara.
- Şimşek, B., 2016. İris Karşılaştırma Algoritmasının OpenCL Dili Kullanılarak Hızlandırılması, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 81s, Ankara.
- Tay, R., 2013. OpenCL Parallel Programming Development Cookbook. Packt Publishing, 286s, Birmingham.
- Tonguç, G., 2007. Görüntü İşleme Teknikleri Kullanılarak Meyve Tasnifi. Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 90s, Isparta.

Tosun, O., 2015. Görüntü İşleme ile Yaprak Alanı Ölçüm Sistemi Tasarımı. Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, YL Tezi, 70s, Isparta.

Wu, Q., Ha, Y., Kumar, A., Luo, S., Li, A. & Mohamed, S. 2014. A Heterogeneous Platform with GPU and FPGA for Power Efficient High Performance Computing. College of Information Science & Engineering, Hunan University, Changsha, China. 220-223.