

## Uzaktan Algılamada Rutin İşlemlerin Otomasyonu: Görüntülerin Sadece Veri ile İlgili Olan Kısımlarının Kesimi

(Automation of Routine Tasks in Remote Sensing: Extracting Only Useful Parts of Imagery)

Ali Levent YAĞCI, Ahmet ÇAM

Harita Genel Komutanlığı, Fotogrametri Dairesi, Uzaktan Algılama Şubesi, Ankara  
yagciali2002@gmail.com, ahmet.cam@hgk.msb.gov.tr

### ÖZ

Amerika Birleşik Devletleri'ndeki (ABD) Ulusal Havacılık ve Uzay Dairesi (National Aeronautics and Space Administration; NASA), Jeolojik Araştırmalar Merkezi (United States Geological Survey; USGS) ve Ulusal Okyanus ve Atmosfer Dairesi (National Oceanic and Atmospheric Administration; NOAA) gibi uluslararası araştırma ve veri üretim/dağıtım merkezlerinde, hava ve uydu görüntülerinin ön işlemleri büyük bir ölçüde insandan bağımsız bir şekilde gerçekleştirilmektedir. Bu işlemler genellikle C veya Python gibi bilgisayar programlama dillerinde geliştirilen programlar sayesinde, baştan sona otomatik olarak bilgisayarlar tarafından gerçekleştirilmektedir. Bu çalışmalara paralel olarak, Harita Genel Komutanlığı'nda (HGK), hava ve uydu görüntüleri ile ilgili rutin işlemlerin, kullanıcılardan bağımsız olarak otomatikleştirme çalışmalarına başlanmıştır. HGK'nin sorumlukları içinde olan, hava ve uydu görüntülerinin kenarında bulunan ve veri ile ilgili olmayan siyah kısımlarının otomatik olarak kesilmesi için Python dilinde bir program geliştirilmiştir. Ayrıca, Python dilini bilmeyen kullanıcıların da bu programı kullanabilmesi için, bu programa ESRI ArcGIS yazılımı çatısı altında çalışan grafik kullanıcı arabirimi de geliştirilmiştir. Böylece, işlemlerin kullanıcıdan bağımsız otomasyonu sayesinde zaman ve insan gücü tasarrufu yapılacaktır. Ayrıca, görüntü kesim işlemleri belli bir standarda oturtulacak ve kullanıcı hataları en aza indirgenecektir.

**Anahtar Kelimeler:** Uzaktan Algılama, Görüntü İşleme, Otomasyon, Python, ArcGIS

### ABSTRACT

In the international research and data production/dissemination centers such as National Aeronautics and Space Administration (NASA), United States Geological Survey (USGS) and National Oceanic and Atmospheric Administration (NOAA), preprocessing of aerial and satellite imagery is generally carried out without human interaction. Computer programs written in C, Python or other programming languages are utilized to process imagery in an automated fashion from start to end. Automation of routine imagery tasks at the General Command of Mapping has been recently begun to process aerial and remote sensing images without any human interaction. A stand-alone python script has been developed to clip imagery in an automated fashion. Furthermore, this script is converted to a tool with graphical user interface in ESRI ArcGIS software so that users who are not familiar with Python programming language can use it. Therefore, elimination of human interaction with software in data

processing tasks can help save time and human effort. Also, automation will help reduce unintended user errors by standardizing image processing tasks.

**Keywords:** Remote Sensing, Image Processing, Automation, Python, ArcGIS

### 1. GİRİŞ

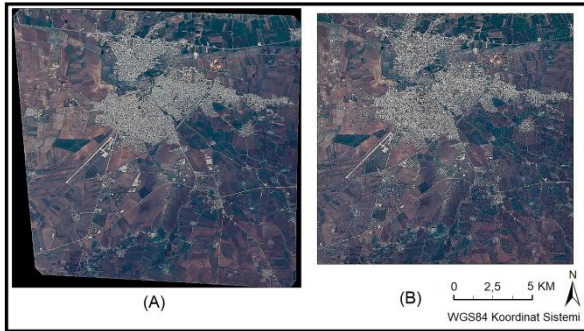
Milli Savunma Bakanlığına (MSB) bağlı HGK kalkınma, savunma, güvenlik amacıyla ve istihbarat faaliyetlerinde kullanılmak üzere rutin olarak Türkiye içi güncel yüksek çözünürlükle ( $\approx 50$  cm) hava görüntüleri toplamak ve üretmektedir (URL-1). Ayrıca, havadan görüntüsü alınamayan bölgelerde ise, Pleiades, GeoEye-1 ve Göktürk-1 gibi uydulardan toplanmış yüksek çözünürlüklü uydu görüntülerini kullanmaktadır. Bu güncel hava ve uydu görüntüleri daha sonra, HGK tarafından Coğrafi Analiz Sistemi'nde (CAS), diğer bir deyişle Sanal Küre'de, Türk Silahlı Kuvvetleri'nin (TSK) kullanımına hazır duruma getirilmektedir. Ayrıca, istek olduğunda bu veriler kamu kurum ve kuruluşlarına da dağıtılmaktadır.

HGK tarafından toplanan hava görüntülerinin tüm ön işlemleri (preprocessing) HGK'da yapılmaktadır. İlk aşamada, elektromanyetik spektrum'un kırmızı, yeşil, mavi, yakın kızıl ötesi ve pankromatik dalga boylarında toplanan ham verilerin kalibrasyonu yapılmaktadır ve renklerinin dengelenip birleştirilerek görüntü haline getirilmektedir. Daha sonra, bu görüntüler yeryüzündeki yer kontrol noktaları yardımıyla dengelenir. Sonra, bu görüntüler kamera parametreleri, hassas dış yöneltme parametreleri ve sayısal arazi modeli yardımı ile orto-görüntü haline getirilmektedir (URL-2). En son olarak, bu görüntüler CAS'a yüklenmeden önce görüntü kenarında bulunan ve görüntü ile ilgili olmayan alanlar kesilmektedir ve kesilen görüntü ECW (Enhanced Wavelet Compression) formatında kaydedilmektedir.

HGK'da görüntü işleme, genellikle ArcGIS, ERDAS, GLOBAL Mapper gibi yazılımlar yardımıyla yapılmaktadır. Bu işlemler, uydu ve hava görüntülerinin boyutlarına (görüntülerin bilgisayarda GB biriminde kapladığı alan) paralel olarak uzun sürmektedir. Ayrıca, görüntü işleme

kullanıcılar tarafından manuel yapıldığı için sadece mesai saatleri içinde gerçekleştirilmektedir. İstisnai olarak, gerekli görüldüğünde, görüntü birleştirme (mosaic) gibi çok zaman alan işlemler, mesai bitimine yakın başlatılmaktadır. Ama otomasyon olmadığı için, mesai saatleri dışında verilen bir iş yalnızca tek bir işlem veya görüntü ile sınırlıdır.

Güncel hava ve uydu görüntüleri CAS'a yüklemeye önce, görüntü kenarlarında bulunan, Şekil 1-A'da gösterilen veri olmayan kısımlar (siyah renkli bölümler) kesilerek Şekil 1-B'deki görüntü haline getirilerek yüklenmektedir. Birbirine komşu olan görüntülerdeki ortak alanlar yüzünden, sistemde görüntüler üst üste binmektedir. Bir görüntüdeki veri olmayan alanlar (siyah renkli bölgeler), komşu görüntülerdeki veri olan alanların üstüne bindiği için, komşu görüntülerdeki veri olan alanlar görüntülenememektedir. Bu nedenle bu siyah kısımların kesilmesi gerekmektedir. Bu işlem manuel olarak, siyah bölgeler dışında 4 köşe noktası seçilmek suretiyle teker teker mesai saatlerinde yapılmaktadır. Görüntünün boyutlarına göre her bir görüntünün kesimi yaklaşık 30 dakika ile 1 saat arasında gerçekleştirilmektedir.



Şekil 1. Programa giren (A) ve programdan çıkan (B) 201602040752350 adlı uydu görüntüsü. Bu görüntünün diğer özellikleri Tablo 1' de mevcuttur.

Amerika Birleşik Devletleri'ndeki (ABD) Ulusal Havacılık ve Uzay Dairesi (National Aeronautics and Space Administration; NASA), Jeolojik Araştırmalar Merkezi (United States Geological Survey; USGS) ve Ulusal Okyanus ve Atmosfer Dairesi (National Oceanic and Atmospheric Administration; NOAA) gibi uluslararası araştırma ve veri üretim/dağıtım merkezlerinde, hava ve uydu görüntülerinin ön işlemleri büyük bir ölçüde otomatik bir şekilde yapılmaktadır (URL-3; URL-4 ; URL-5). Bilgisayarlar, bu işlemleri C veya Python gibi bilgisayar programlama dillerinde yazılan programlar vasıtasıyla insandan bağımsız

gerçekleştirir. Bilhassa Python programlama dili Coğrafi Bilgi Sistemleri (CBS) ve uzaktan algılama alanlarında çok yoğun olarak kullanılmaktadır. Örneğin, ArcGIS yazılımının resmi dili python programlama dilidir. Ayrıca, Python dilinde yazılmış çeşitli modeller de bulunmaktadır. Bu modellere örnek olarak, uydu görüntülerinden kuraklığın belirlenmesi ve yeryüzünden (bitkilerden ve topraktan) su buharlaşma miktarının hesaplanması verilebilir (Yagci vd., 2017; Yagci vd., 2015; Deng vd., 2013).

Bu çalışmalara paralel olarak, HGK'da, hava ve uydu görüntüleri ile ilgili rutin işlemlerin insandan bağımsız olarak otomatikleştirme çalışmalarına başlanmıştır. Bu çalışmada, görüntülerin veri ile ilgili olmayan siyah kısımlarının otomatik olarak nasıl kesileceğini ve bu gibi rutin görüntü işlemlerde nelere dikkat etmek gerektiğini anlatılacaktır. Ayrıca, kullanılan yöntem sadece bu uygulama ile sınırlı olmadığı için, ileriye dönük bu yöntemin diğer uygulamalarda nasıl kullanılabileceğinden bahsedilecektir.

## 2. VERİLER

Yazılan programın test edilmesi amacıyla çeşitli boyutlarda 18 tane uydu görüntüsü seçilmiştir (Tablo 1). Bu görüntüler, program defalarca değişik parametreler ile çalıştırılarak kesilmiştir ve her sefer programın çalışma süresi kayıt edilmiştir.

Tablo 1. Çalışmada kullanılan görüntülerin özellikleri.

Görüntü Adı	Sütun Sayısı	Satır Sayısı	Boyut (GB)	
			1	2
201512080838114	42648	34327	8,18	1,63
201602040752350	36593	35418	7,24	1,39
201602060826344	51856	19181	5,56	1,09
201604010802398	51298	14034	4,02	0,79
201609300804329	49184	20572	5,65	1,13
201701050807586	56130	64836	20,34	4,05
201701060759145	52225	78601	22,94	4,57
201701210833195	55996	24830	7,77	1,38
201702240822210	55691	105557	32,85	6,47
201703160818341	52368	83318	24,38	4,85
201703190846230	53746	45809	13,76	2,67
201703230815038	53641	41286	12,38	2,47
201704040822100	55861	47635	14,87	2,85
201704050814383	52786	132070	38,96	7,76
201704100826060	56438	201499	63,55	12,50
201704270755543	58185	45278	14,72	2,87
201705050834185	52076	38910	11,32	2,26
201705060826176	41565	81653	18,96	3,73

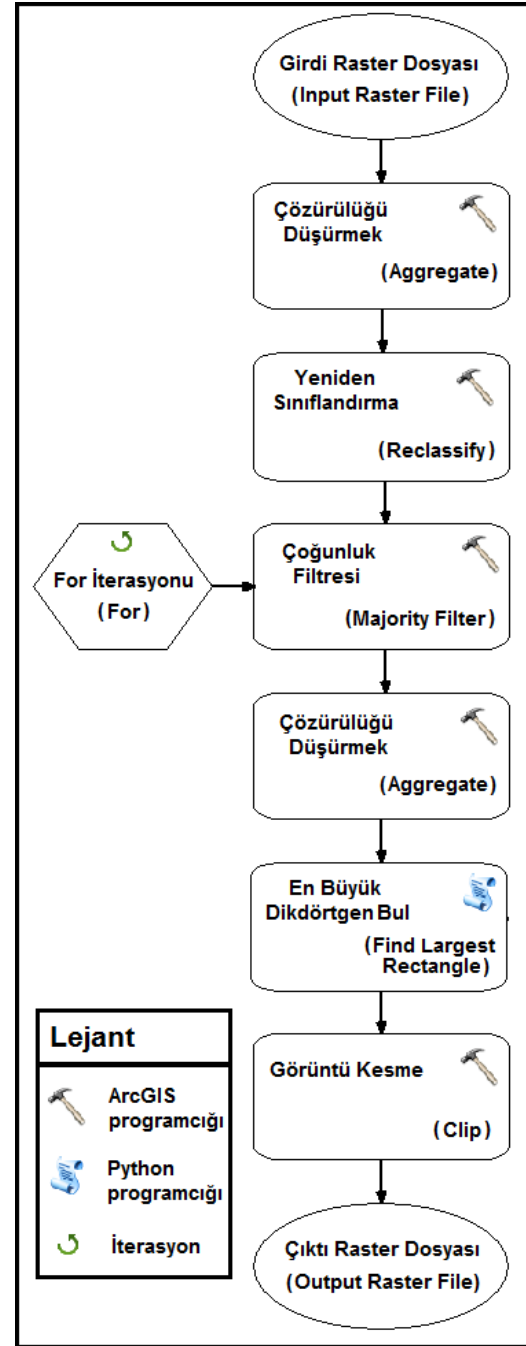
Her bir görüntü, elektromanyetik spektrumun kırmızı, yeşil ve mavi bölgelerinde toplanmış 3 bant içermektedir ve JPEG2000 (uzantısı: jp2) kayıplı (lossy) sıkıştırma formatında, HGK'ya teslim edilmektedir. Bu format, dalgacık (wavelet) teknolojisi kullanarak, görüntüyü kullanıcının belirleyeceği sıkıştırma kalitesine göre sıkıştırılmaktadır (URL-6). Örneğin, Tablo 1'de gösterilen 18 tane görüntünün boyutu (Boyut 1) JPEG2000 formatıyla sıkıştırıldıktan sonra (Boyut 2) ortalama %80 azalmıştır. Bu nedenle, görüntülerin %80 sıkıştırma kalitesi ile sıkıştırıldığı ve yaklaşık %20 veri kaybı yaşandığı söylenebilir. Özetle, bu format ile görüntülerin boyutu yüksek derecede azaltılmasına rağmen, görüntüler sıkıştırıldıktan sonra görüntülerde veri kaybı yaşanmaktadır.

### 3. METOTLAR

ArcGIS, ESRI şirketi tarafından geliştirilen temelde lisanslı bir CBS yazılımıdır ancak içindeki programlar uzaktan algılama görüntülerini de işleyebilmektedir. Bu yazılımı, diğer CBS ve uzaktan algılama yazılımlarından ayıran önemli özelliği ise Python programlama dili ile uyumlu çalışabilmesidir. Bu sayede, bu yazılımın hem günlük görüntü işleme hem de bilimsel çalışmalarda kullanımına olanak sağlamaktadır.

Görüntü işleme, bu yazılımın ArcToolbox kısmında bulunan programcıklar (tool) vasıtasıyla gerçekleştirilebilir. Ayrıca, ArcToolbox'da bulunmayan işlemler için Python dilinde yazılan kodlar vasıtasıyla görüntü işleme gerçekleştirilebilir. ArcGIS Python ile uyumlu çalıştığı için Python dilinde yazılan kodlar (script), ArcToolbox'a programcık olarak eklenebilmektedir ve bu diğer mevcut programcıklar ile uyumlu çalışabilmektedir. Örneğin, bu çalışma için yazılan görüntü kesme programı, hem mevcut programcıklardan hem de Python dilinde yazılan programcıkların birleştirilmesiyle oluşturulmuştur. Programda kullanılan programcıkların gösterildiği diagram Şekil 2'de verilmiştir.

ArcToolbox'da bulunan programcıklar hem kendi başına hem de başka bir programcığın veya modelin bir parçası olarak çalışabilmektedir. Buna ilaveten, bu programcıklar, ArcPy modülü sayesinde Python kodları içinde de kullanılabilir. ArcPy, ArcToolbox'da bulunan programcıkların Python içinde ArcGIS programı çalıştırılmadan kullanılabilmesini sağlamaktadır.



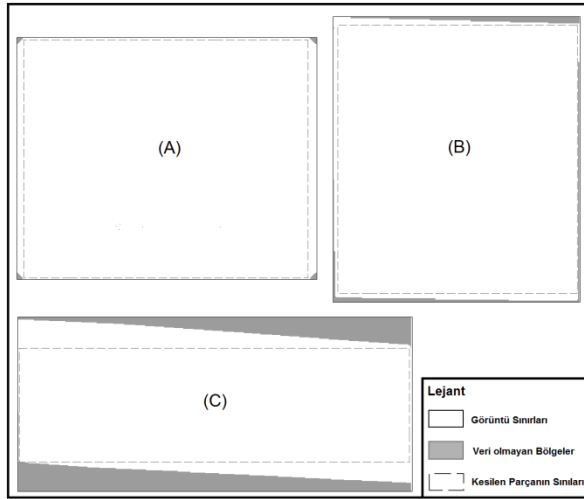
Şekil 2. Otomatik görüntü kesme programcığı işlem akış diyagramı

#### a. Kullanılan Programcıklar

Genelde, hava ve uydu görüntüleri bilgisayarda büyük bir yer kaplamaktadır. Örneğin, bu çalışmada yazılan programın testi için kullanılan görüntülerin sıkıştırılmamış boyutları 4 GB ila 64 GB arasındadır (Tablo 1). Bu nedenle, görüntü kesme işleminin hızlı ve sorunsuz gerçekleştirilebilmesi için çözünürlüğünü düşürmek gerekmektedir.

Görüntü kesme programcığının ilk aşamasında görüntünün çözünürlüğü düşürülmektedir. Bu çalışmada düşük, düşük-orta, büyük-orta ve büyük boyutlu görüntülerin sırasıyla 50, 100, 250 ve 500 kat çözünürlüğü düşürülmektedir.

İkinci aşamada, görüntü piksel değerleri yeniden sınıflandırılmaktadır (bu işlem görüntü sınıflandırma ile karıştırılmamalıdır). Veri içeren pikseller bir sınıfta (piksel değeri 1), veri içermeyen pikseller (piksel değeri 0) başka bir sınıf olmak üzere toplam iki sınıfta toplanmaktadır ve sadece iki değer (binary) bulunan yeni bir görüntü oluşturulmaktadır. Şekil 3'te bu sınıflandırmadan oluşan iki sınıflı görüntüler gösterilmiştir.

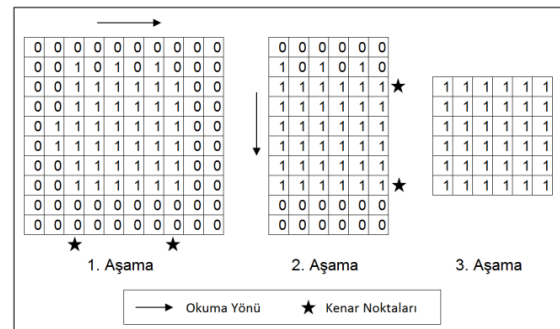


Şekil 3. Program ile kesilen 201512080838114 (A), 201701050807586 (B) ve 201701210833195 (C) adlı görüntülerin orijinal sınırları, veri olmayan kısımlarının dağılımı ve kesilen parçaların sınırları.

Görüntülerin veri ile ilgili olmayan kısımları genellikle görüntülerin kenarlarında bulunmaktadır (Şekil 1-A; Şekil 3-A, B ve C). Ama bazı görüntülerde, görüntülerin orta kısımlarında sıfır (0) değerine sahip az sayıda siyah piksel veya piksel grupları görülmüştür. Bu görüntünün ortasında olan sıfır değerine sahip veri içermeyen kısımlar, "en büyük dikdörtgen bul" programcığının istenilen sonuçlar vermesini engellemektedir. Bu nedenle, 3. aşamada yeniden sınıflandırılan görüntü, 3 kez iterasyon ile 3x3 boyutlu çoğunluk filtresinden geçirilir. Böylelikle, görüntü kenarlarında çoğunluk olan sıfır pikselleri (siyah) korunur ama görüntü ortasında olan sıfır pikselleri bir pikselleri ile yer değiştirir.

Bir sonraki aşamada, görüntünün çözünürlüğü 2 kat daha düşürülür. İlk aşamada da, görüntü çözünürlüğü düşürülmektedir ama bazı görüntülerin kenarlarında çok küçük siyah bölümler bulunmaktadır. Eğer bu görüntülerin çözünürlüğü çok düşürülürse bu sefer bu siyah bölümler kaybolmaktadır ve program görüntülerin kenarında siyah bölgeler yok diye algılayıp kesme işlemi gerçekleştirmemektedir. Onun için çözünürlüğün kademeli olarak iki aşamada düşürülmesi daha uygundur.

Kesme işleminde bir önceki aşamada, "en büyük dikdörtgen bul" programcığı yardımıyla siyah bölgelerden arındırılmış en büyük parçanın dört köşe koordinatları bulunur. Bu aşamada, çözünürlüğü düşürülmüş 0 ve 1 piksel değerlerinden oluşan görüntü, bir diziye (array) yazdırılır. Bu dizi önce soldan sağa ve yukarıdan aşağıya okunarak, her satır ve sütun için siyah bölgelerin bittiği ve yeniden başladığı piksel numaraları bulunur. İlk aşamada, dizi soldan sağa okunur, siyah bölgelerin başladığı satır piksel numaralarının en büyüğü, siyah bölgelerin bittiği satır numaralarının en küçüğü seçilir ve bu numaralara göre dizi kesilir. İkinci aşamada, kesilen dizi yukarıdan aşağıya okunur ve siyah bölgelerin başladığı sütun piksel numaralarının en büyüğü, siyah bölgelerin bittiği sütun numaralarının en küçüğü seçilir ve dizi tekrar kesilir. Böylelikle, dizi 0 değerli piksellerden tamamen arındırılmış olur. Kesim işleminin algoritması görsel olarak Şekil 4'te anlatılmıştır. Bu işlemler 3 kez daha yapılır ve her seferinde dizi 90° döndürülür. Her görüntüden toplamda 4 tane kesilmiş dizi bulunur. Alan hesabına göre bu dizilerin en büyük olanı bulunur. En son olarak, en büyük alanlı dizinin 4 köşe noktalarının koordinatları bulunur. Bu aşama, bu programın en önemli kısmıdır ve diğer programcıkların aksine ArcGIS ArcToolbox'da bulunmamaktadır. Bu programcık, sıfırdan Python dilinde yazılmıştır ve en kısa süren kısmıdır. Diğer kullanılan programcıklar ArcToolbox'da mevcuttur.



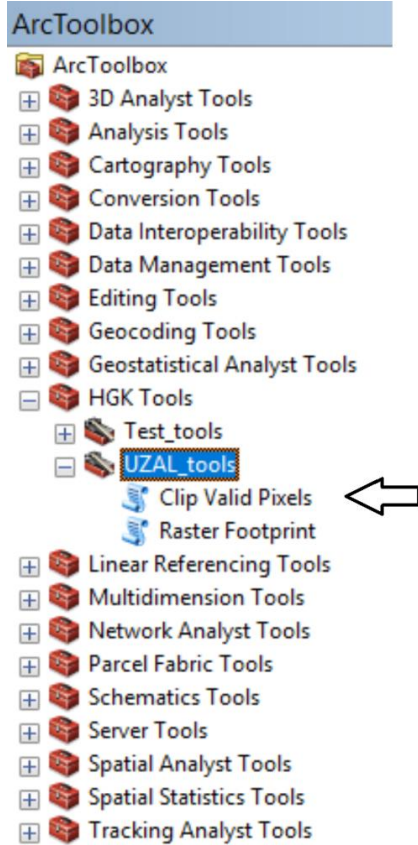
Şekil 4. "En büyük dikdörtgen bul" programcığının algoritması

Programın en son aşamasında, bir önceki aşamada bulunan köşe koordinatlarına göre görüntü kesilir. Bu kısım programın en uzun süren kısmıdır.

Ayrıca, program “en büyük dikdörtgen bul” aşamasından önce, görüntüde siyah bölge var mı diye kontrol etmektedir. Eğer görüntüde siyah bölge yoksa program o aşamada durur ve girdi görüntüsünü kesmeden çıktı klasörüne kopyalar.

## b. Programcığın Kullanımı

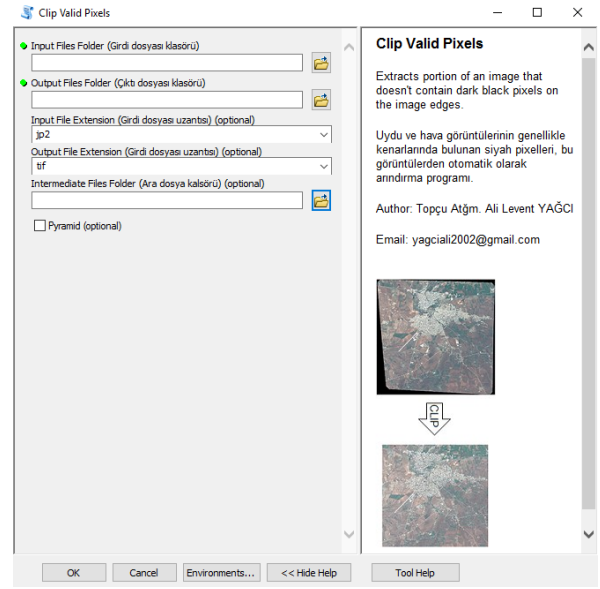
Python dilinde yazılan kod, ArcGIS programı açılmadan Python komut satırından çalıştırılabilir. Ama bu işlem için basit Python dili bilgisi gereklidir. Bu yüzden, görüntü kesme programcığına grafik kullanıcı arabirimi eklenerek ArcToolbox’da sunulmuştur (Şekil 5). Böylelikle, kullanıcıların herhangi bir programlama dili bilmesine gerek olmadan ve ArcToolbox’daki diğer programcıkları kullandıkları şekilde, bu programcığı da kullanılabilir olacaktır.



Şekil 5. Görüntü Kesme programcığının ArcGIS ArcToolbox’daki görünümü

Programcığın grafik kullanıcı arabirimi Şekil 6’daki gibidir. Programda 2’si zorunlu 4’ü

opsiyonel olacak şekilde toplam 6 tane parametre vardır. Girdi dosyalarının bulunduğu klasör ile kesilen görüntülerin yazılacağı klasör zorunlu parametrelerdir. Girdi dosyalarının uzantısı, kesilen dosyaların hangi uzantıda yazılacağı, programın çalışması sırasında oluşan ara dosyaların yazılacağı klasör ve kesilen görüntünün piramitlerinin oluşturulup oluşturulmayacağı opsiyonel parametrelerdir.



Şekil 6. Programcığın grafik kullanıcı arabirimi

Uydu görüntüleri, JPEG2000 (dosya uzantısı: jp2) formatı ile teslim edildiği için, girdi dosyaları parametresi jp2 olarak önceden tanımlanmıştır. Çıktı dosyaları ise GeoTIFF (dosya uzantısı: tif) formatı olarak programa önceden tanımlanmıştır. Ayrıca program kesilen görüntünün piramitlerini oluşturmayacak ve ara dosyaları dışarıya yazmayacak şekilde önceden ayarlanmıştır. Örnek olarak, program görüntüyü istenildiği gibi kesmediğinde (Görüntü kesme sonrasında oluşan görüntüde veri ile ilgili olmayan siyah renkli pikseller kalmışsa) veya programın çalışması sırasında bir hata olduğunda, ara dosyaları belli bir klasöre yazdırılabilir. Bu ara dosyalar incelenerek, sorunun hangi aşamadan kaynaklandığı bulunabilir. Çıktı dosyaları ve piramit parametrelerin neden yukarıda bahsedilen değişkenler ile programa önceden tanımlandığı “bulgular” kısmında anlatılacaktır.

## 4. BULGULAR

Yazılan görüntü kesme programcığı, Tablo 1’de verilen 18 tane görüntü ile defalarca test edilmiştir. Bu programcık, görüntü başına en az

3'er kez olmak üzere çeşitli opsiyonel parametrelerle hem Python hem de ArcGIS ortamında çalıştırılmıştır ve programcık her çalıştırıldığında geçen süreler kaydedilmiştir. Bu bölümde, bu çalışma sürelerinin istatistikleri sunulacaktır.

#### a. Görüntü Kesme Süresi

Görüntü kesme programcığının her bir aşamasının ne kadar sürdüğü ve bu aşamaların toplamı Tablo 2'de verilmiştir. Bu programcık Python ortamında piramit seçeneği kapalı olarak çalıştırılmıştır ve kesilen dosyalar GeoTIFF formatında kaydedilmiştir. Görüntü kesme işleminin yapıldığı bilgisayarın özellikleri Tablo 3'de listelenmiştir.

Programcık en kısa 5,7 dakika çalışarak en küçük boyutlu "201604010802398" adlı görüntüyü ve en uzun 95,3 dakika çalışarak en büyük boyutlu 201704100826060 adlı görüntüyü

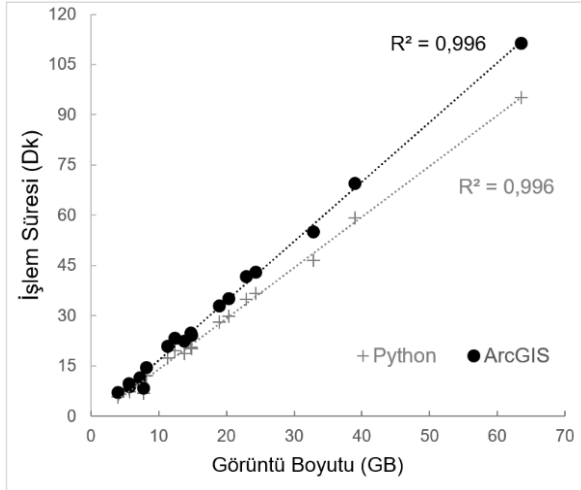
otomatik olarak kesmektedir (Tablo 2). Görüntü kesme süresi, görüntü boyutu ile doğru orantılıdır (Şekil 7). Diğer bir deyişle, görüntünün boyutu büyüdükçe görüntüyü kesme süresi de uzamaktadır.

Tablo 2. Görüntü kesme işleminin yapıldığı bilgisayarın özellikleri

İşlemci Marka ve Model	Intel Xeon X5647
İşlemci Hızı (GHz)	2,93
İşlemci Çekirdek Sayısı	4
Yüklü Bellek (GB)	16
Sistem Sabit Disk (GB)	256
Depolama Sabit Disk (TB)	4
İşletim Sistemi	Windows 10 Enterprise
İşletim Sistemi Türü (bit)	64

Tablo 3. Görüntü kesme işleminde ortalama çalışma süreleri. Programcık Python ortamında piramit seçeneği kapalı olarak çalıştırılmıştır ve kesilen dosyalar GeoTIFF formatında kaydedilmiştir.

Görüntü Adı	Çözünürlüğü Düşürmek (Dk.)	Yeniden Sınıflandırma (Dk.)	Çoğunluk Filtresi (Dk.)	Çözünürlüğü Düşürmek (Dk.)	En Büyük Dikdörtgen Bul (Dk.)	Görüntü Kesme (Dk.)	Toplam Zaman (Dk.)
201512080838114	1,750	0,009	0,033	0,010	0,002	10,361	<b>12,168</b>
201602040752350	1,702	0,010	0,035	0,012	0,002	7,672	<b>9,432</b>
201602060826344	1,312	0,010	0,034	0,009	0,003	6,711	<b>8,081</b>
201604010802398	0,903	0,009	0,034	0,012	0,003	4,812	<b>5,775</b>
201609300804329	1,325	0,010	0,035	0,012	0,003	5,903	<b>7,289</b>
201701050807586	4,082	0,011	0,033	0,013	0,002	25,669	<b>29,811</b>
201701060759145	5,052	0,010	0,035	0,011	0,003	29,843	<b>34,958</b>
201701210833195	1,693	0,011	0,036	0,014	0,003	5,176	<b>6,933</b>
201702240822210	7,778	0,009	0,036	0,011	0,001	38,749	<b>46,588</b>
201703160818341	5,881	0,010	0,035	0,011	0,003	30,769	<b>36,712</b>
201703190846230	3,023	0,013	0,039	0,014	0,002	15,556	<b>18,647</b>
201703230815038	3,106	0,012	0,038	0,014	0,003	16,374	<b>19,548</b>
201704040822100	3,480	0,008	0,034	0,011	0,003	16,639	<b>20,178</b>
201704050814383	9,115	0,009	0,036	0,012	0,002	49,976	<b>59,151</b>
201704100826060	14,705	0,009	0,036	0,012	0,003	80,306	<b>95,072</b>
201704270755543	3,465	0,013	0,040	0,014	0,003	17,127	<b>20,663</b>
201705050834185	2,654	0,013	0,038	0,015	0,003	14,712	<b>17,434</b>
201705060826176	4,410	0,008	0,032	0,011	0,001	23,572	<b>28,036</b>



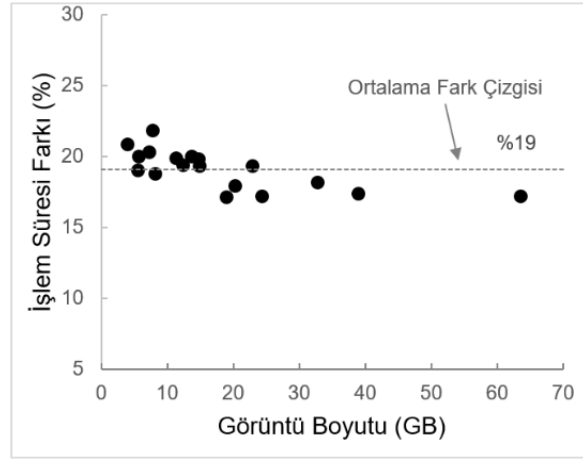
Şekil 7. Görüntü kesme işleminin hem Python (+) hem de ArcGIS (●) ortamında süresinin görüntü boyutları ile ilişkisi.

En uzun aşama görüntü kesme (clip) (Şekil 2) aşamasıdır ve bu aşama ortalama toplam zamanın yaklaşık olarak %83'ünü almaktadır (Tablo 2). İlk aşama olan görüntünün çözünürlüğü düşürülmesi (aggregate) (Şekil 2) ise ikinci uzun aşamadır ve bu aşama ortalama toplam zamanın yaklaşık olarak %16,2'sine tekabül etmektedir (Tablo 2). Geri kalan diğer aşamalar, toplam görüntü kesme süresinin yaklaşık %0,4'ünü almaktadır. En kısa aşama ise en büyük dikdörtgen bul aşamasıdır ve ortalama sadece 0,15 saniye sürmektedir (Tablo 2).

### b. Python- ArcGIS Karşılaştırması

Görüntü kesme işleminin hangi ortamda daha hızlı olduğunu belirlemek için 18 tane görüntü hem yazılan kod ile Python ortamında ve hem de bu kodun grafik kullanıcı arabirimli programcığı ile ArcGIS ortamında her görüntü en az 3 kez olmak üzere kesilmiştir. Her iki ortamdaki işlem süreleri ve bu işlem sürelerinin görüntü boyutları ile olan ilişkileri Şekil 7'de gösterilmiştir. Ayrıca işlem süreleri arasında yüzdesel farklar her görüntü için Şekil 8'de verilmiştir.

Şekil 7'ye göre aynı Python ortamındaki gibi, ArcGIS ortamındaki görüntü kesme süresi görüntü büyüdükçe uzamaktadır. ArcGIS ortamında görüntü kesme süresi Python ortamındaki kesme süresine göre görüntü boyutlarından bağımsız ortalama %19 fazladır (Şekil 8). Bir diğer deyişle, Python ortamında görüntüleri kesmek ortalama %19 zaman kazandırmaktadır.



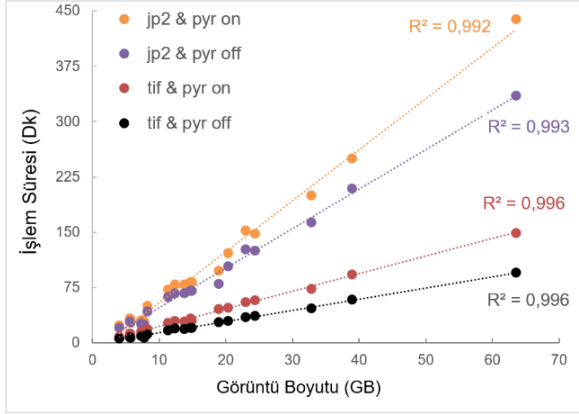
Şekil 8. ArcGIS ortamındaki görüntü kesme süresinin Python ortamındaki görüntü kesme süresine göre farkı. Farklar yüzdesel olarak 18 görüntü için ayrı ayrı hesaplanmıştır.

### c. Parametre Seçimi

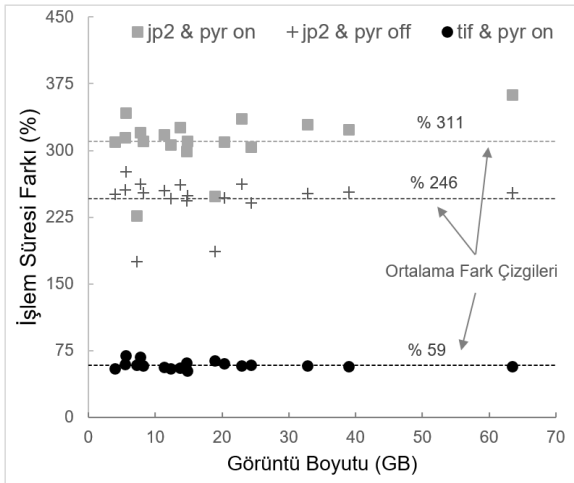
Metotlar kısmında, grafik kullanıcı arayüzlü programcığıyla kesilen görüntülerin GeoTIFF formatı ile piramitler seçeneği kapatılarak bilgisayara yazdırıldığından bahsetmiştik. Bunun nedeni ise Şekil 9 ve 10'da betimlenmiştir. Şekil 9'a göre programcık en hızlı şekilde piramit seçeneği kapatılıp ve çıktı dosyası GeoTIFF seçildiğinde çalışmaktadır. Kesilen dosyayı GeoTIFF formatı ile yazdırırken piramit seçeneğini açmak işlem süresini ortalama %59 uzatmaktadır (Şekil 10). Eğer, piramit seçeneği kapalı iken kesilen görüntü JPEG2000 formatında (sıkıştırma kalitesi %75) sıkıştırılmak istenirse görüntü kesme süresi, ortalama %246 uzamaktadır (Şekil 10). Piramit seçeneği açıkken kesilen görüntüyü JPEG2000 formatıyla yazdırılmak istenirse görüntü kesme süresi ortalama %311 uzamaktadır. Örneğin, programcık Python ortamında, 201512080838114 adlı görüntüyü piramitsiz ve GeoTIFF formatı ile kesmek için 12,2 dakika çalışmaktadır (Tablo 2). Aynı görüntüyü, piramitli GeoTIFF formatı ile kesmek 19,2 dakika, piramitsiz JPEG2000 formatı ile kesmek 42,9 dakika ve piramitli JPEG2000 formatı ile kesmek 49,9 dakika almaktadır. Ayrıca, görüntü kesme süresi görüntünün boyutu ile doğru orantılıdır (Şekil 9).

Kesilen görüntü GeoTIFF formatında, LZW sıkıştırması kullanılarak kayıpsız (lossless) sıkıştırılabilmektedir ve JPEG2000 formatındaki gibi bir sıkıştırma kalitesi seçilememektedir. JPEG2000 formatında ise veri kayıp oranı seçilebilir. Programcıkta, ortam seçeneklerinde

önceden tanımlı %75 sıkıştırma kalitesi kullanılmaktadır ve yaklaşık kesilen görüntünün boyutu % 75 civarı küçültülebilmektedir. Buna karşın, GeoTIFF formatıyla kaydedilen görüntüler LZW ile sıkıştırılınca, görüntüler ortalama %30 küçültülebilmektedir.



Şekil 9. Görüntü kesme programcığının çıktı dosya formatı ile piramit seçeneği değiştirildiğinde oluşan işlem süreleri ve bu sürelerin görüntü boyutları ile olan ilişkisi.

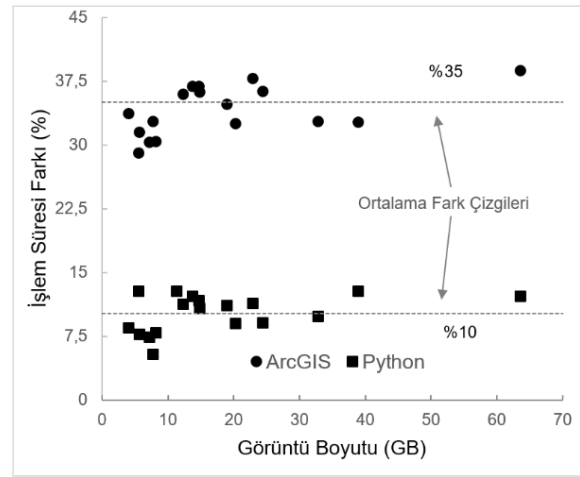


Şekil 10. Görüntü kesme programcığının piramitli GeoTIFF (●), piramitsiz JPEG2000 (+) ve piramitli JPEG2000 formatı (■) ile çalıştırıldığında oluşan işlem sürelerinin, piramitsiz GeoTIFF formatı ile çalıştırıldığında oluşan işlem süresinden farkları.

### ç. Ağ Üzerinden Görüntü Kesme

Görüntüler, merkezi sunucularda muhafaza edilmektedir. Erişim izni olan bütün kullanıcılar bu görüntülere istedikleri zaman ulaşabilmektedirler. Şu ana kadarki görüntü kesme işlemleri, görüntüler kişisel bilgisayara sunucudan kopyalandıktan sonra bu bilgisayar üzerinde gerçekleştirilmiştir. Buna karşın, kullanıcılar

görüntü kesim işlemini, görüntüleri kişisel bilgisayarlarına transfer etmeden gerçekleştirmek istemektedir. Bu nedenle, görüntü kesme işlemi, ağ üzerinden hem Python hem de ArcGIS ortamında gerçekleştirilmiştir ve Python ortamında kesim süreleri Tablo 4'te ve ArcGIS ortamdakiler ise Tablo 5'te verilmiştir. Bu işlemde, görüntüler sunucudan üzerinden okutulup, sunucuya yazılmıştır. Daha sonra bu süreler, aynı Şekil 8 ve 10'daki gibi kişisel bilgisayarda Python ortamında gerçekleştirilen görüntü kesme süreleri (Tablo 2) ile karşılaştırılmıştır (Şekil 11).



Şekil 11. Görüntü kesme programcığının ArcGIS (●) ve Python (■) ortamında merkezi sunucu (server) bilgisayar üzerinden görüntüleri okuyup tekrar aynı yere yazdığına oluşan işlem sürelerinin, kişisel bilgisayar üzerinde Python ortamındaki görüntü kesme süresine göre farkı (Tablo 2).

Şekil 11'e göre, Python ortamındaki görüntü kesme işlem süresi ortalama %10 ve ArcGIS ortamında %35 artmıştır. Toplam 18 görüntünün kesim süresi 476 dakikadan (7 saat 56 dakika) Python ortamında 529 dakikaya (8 saat 49 dakika) ve ArcGIS ortamında ise 648 dakikaya (10 saat 48 dakika) çıkmıştır.

Tablo 1'de verilen 18 görüntünün toplamı yaklaşık 65 GB'tır ve sunucudaki bilgisayar ile kişisel bilgisayar arasındaki dosya transfer hızı ise 80 ile 120 MB/saniye arasında dalgalanmaktadır. Bu hız dosya transferinde gözlemlenen hızdır, bu yüzden sunucuyu kullanan kişi sayısına göre değişiklik gösterebilir. Dosyaların tek yönlü sunucudan kişisel bilgisayara taşınması 10-11 dakika almaktadır. Görüntüler kesildikten sonra GeoTIFF formatından ECW formatına dönüştürülen görüntülerin boyutu daha da azalacağından, görüntülerin sunucuya transferi daha da



kısalacaktır. Böylelikle görüntülerin çift yönlü transferi yaklaşık 20 dakika civarı olacaktır. Sonuç olarak, 18 görüntünün toplam kesim süresi

476 dakikadan (7 saat 56 dakika) 496 dakikaya (8 saat 16 dakika) çıkacaktır ve artış oranı %4 olacaktır.

Tablo 4. Ağ üzerinden gerçekleştirilen görüntü kesme işlemlerinin ortalama çalışma süreleri. Programcık Python ortamında piramit seçeneği kapalı olarak çalıştırılmıştır ve kesilen dosyalar GeoTIFF formatında kaydedilmiştir.

Görüntü Adı	Çözünürlüğü Düşürmek (Dk.)	Yeniden Sınıflandırma (Dk.)	Çoğunluk Filtresi (Dk.)	Çözünürlüğü Düşürmek (Dk.)	En Büyük Dikdörtgen Bul (Dk.)	Görüntü Kesme (Dk.)	Toplam Zaman (Dk.)
201512080838114	2.098	0.010	0.044	0.014	0.001	10.958	13.129
201602040752350	1.748	0.009	0.041	0.014	0.003	8.308	10.126
201602060826344	1.603	0.009	0.043	0.014	0.003	7.443	9.118
201604010802398	0.984	0.010	0.043	0.015	0.003	5.211	6.265
201609300804329	1.410	0.010	0.042	0.014	0.003	6.373	7.853
201701050807586	4.884	0.009	0.042	0.014	0.003	27.547	32.501
201701060759145	5.694	0.010	0.043	0.014	0.001	33.172	38.937
201701210833195	1.748	0.010	0.043	0.014	0.003	5.489	7.308
201702240822210	7.868	0.010	0.043	0.014	0.001	43.248	51.187
201703160818341	5.900	0.010	0.044	0.014	0.003	34.083	40.056
201703190846230	3.139	0.010	0.045	0.014	0.003	17.703	20.915
201703230815038	3.156	0.010	0.043	0.014	0.003	18.524	21.752
201704040822100	3.362	0.010	0.043	0.014	0.003	18.919	22.353
201704050814383	9.461	0.010	0.042	0.014	0.001	57.177	66.707
201704100826060	15.070	0.010	0.044	0.014	0.002	91.562	106.704
201704270755543	3.583	0.009	0.043	0.014	0.002	19.427	23.081
201705050834185	2.852	0.010	0.044	7.007	0.003	16.722	19.645
201705060826176	4.625	0.009	0.042	0.014	0.003	26.450	31.144

Tablo 5. Ağ üzerinden gerçekleştirilen görüntü kesme işlemlerinin ortalama çalışma süreleri. Programcık ArcGIS ortamında piramit seçeneği kapalı olarak çalıştırılmıştır ve kesilen dosyalar GeoTIFF formatında kaydedilmiştir.

Görüntü Adı	Çözünürlüğü Düşürmek (Dk.)	Yeniden Sınıflandırma (Dk.)	Çoğunluk Filtresi (Dk.)	Çözünürlüğü Düşürmek (Dk.)	En Büyük Dikdörtgen Bul (Dk.)	Görüntü Kesme (Dk.)	Toplam Zaman (Dk.)
201512080838114	3.284	0.041	0.122	0.039	0.001	12.385	15.873
201602040752350	2.902	0.044	0.130	0.042	0.001	9.177	12.296
201602060826344	2.219	0.043	0.125	0.040	0.002	8.004	10.432
201604010802398	1.656	0.044	0.127	0.043	0.002	5.852	7.723
201609300804329	2.312	0.039	0.127	0.040	0.001	7.066	9.585
201701050807586	7.993	0.045	0.128	0.040	0.001	31.303	39.509
201701060759145	9.198	0.039	0.119	0.042	0.001	38.799	48.198
201701210833195	2.960	0.040	0.130	0.040	0.001	6.035	9.206
201702240822210	12.936	0.044	0.130	0.041	0.001	48.695	61.848
201703160818341	9.721	0.040	0.126	0.039	0.001	40.116	50.042
201703190846230	5.155	0.044	0.125	0.039	0.002	20.158	25.523
201703230815038	5.011	0.044	0.133	0.041	0.002	21.352	26.582
201704040822100	5.774	0.045	0.133	0.042	0.002	21.495	27.492
201704050814383	14.514	0.045	0.012	0.039	0.001	63.767	78.490
201704100826060	27.357	0.040	0.121	0.039	0.001	104.392	131.950
201704270755543	5.946	0.039	0.122	0.038	0.001	22.136	28.283
201705050834185	4.683	0.045	0.130	0.041	0.002	21.596	26.497
201705060826176	7.471	0.044	0.118	0.038	0.001	30.112	37.784

## 5. SONUÇ VE ÖNERİLER

Bu çalışmada, hava ve uydu görüntülerinin kenar kısmında bulunan veri ile ilgili olmayan siyah kısımlarının, Python programlama dili kullanarak nasıl otomatik kesilebileceği gösterilmiştir. Geliştirilen programcık bu çalışma kapsamında 18 farklı görüntü ile test edilerek anlatılmıştır ama şu ana kadar bu programcık ile 200'den fazla farklı boyutlarda görüntü kesilmiştir. Bu görüntülerin hepsi, programcık tarafından %100 doğrulukla kesilmiştir (görüntülerden veri olmayan kısımlar tamamen arındırılmıştır).

Bu programcık sayesinde gelen görüntüler, kullanıcıya ihtiyaç duyulmadan art arda kesilebilecektir. Bu özellik sayesinde, mesai bitimine yakın başlatılan programcık ile ertesi gün mesai başlangıcına kadar 25-30 görüntü kesilebilecektir. Hafta sonları ise daha fazla görüntü kesilebilmektedir. Böylelikle hem zamandan hem de iş gücünden kazanç oluşacaktır. İlaveten, manuel yöntemde görüntü kullanıcı tarafından kesilirken, kesim alanı göz kararı belirlenmektedir. Bu durumdan dolayı görüntü kenarlarında kalan veri içeren bölümler kesim alanının dışında kalmaktadır ve bu durum veri israfına neden olmaktadır. Programcık sayesinde veri israfı en aza indirgenecek ve görüntü kesme işlemi belli bir standarda oturtulacaktır.

Bu çalışmada uygulanan yöntem genel olup, başka uygulamaların otomatikleştirilmesi için de kullanılabilir. Mesela, aynı yöntem ile görüntülerin yeryüzüne düşen alanlarını otomatik olarak çıkartan bir grafik kullanıcı arabirimli programcık daha geliştirilmiştir (Raster Footprint; Şekil 5) ve bu programcık 5700 üzeri görüntü ile test edilmiştir. Daha sonra, bu alanların birleştirilmesi ile verisi olmayan bölgeler belirlenmiştir.

Bu yöntemde, hem ArcGIS yazılımında bulunan programcıklar hem de yazılımda olmadığı için baştan geliştirilen programcık, Python dilinde büyük bir programcıkta birleştirilmiştir. Sonrada, programlama bilmeyen kullanıcıların kullanması için, ArcGIS yazılımı çatısı altında bu programcığa grafik kullanıcı arabirimi yapılmıştır.

Sonuçlara göre, görüntüler Python ortamında piramit seçeneği kapalı GeoTIFF formatında en hızlı kesilmektedir. Görüntü kesme işlemi aynı seçenekler ile ArcGIS ortamında yaklaşık %19 daha yavaş gerçekleştirilmektedir. Eğer kesilen görüntüler piramitsiz JPEG2000 sıkıştırma formatı seçenekleri ile kesilmek istenirse kesim

süresi %246 artmaktadır. Bu karşın, kesilen görüntüler sıkıştırıldığı için bilgisayarda daha az yer kaplamaktadır. Kesilen görüntüler GeoTIFF ve JPEG2000 formatıyla sırasıyla ortalama %30 ve %75 küçültülebilmektedir. Ayrıca, JPEG2000 formatıyla sıkıştırılma kalitesi yükseltilecek küçültme oranı daha da yükseltilebilir.

Teslim edilen görüntülerin en hızlı biçimde kesilip Sanal Küre sistemine yüklenmesi için, kullanım kolaylığı da göz önünde bulundurularak, görüntülerin ArcGIS ortamında piramit seçeneği kapatılarak GeoTIFF formatıyla kesilmesi en uygundur. Şu anda bu seçenekler programcığa önceden tanıtılmıştır. Kullanıcılar sadece girdi ve çıktı klasörlerini tanıtarak ve diğer parametreleri değiştirmeden görüntü kesme işlemini gerçekleştirebilir. Ayrıca, sunucu üzerinden görüntülerin okunması ve yazılması görüntü kesme işlemini yavaşlatmaktadır. Bu nedenle, görüntülerin önce kişisel bilgisayara taşındıktan sonra, görüntü kesme işleminin bu bilgisayarda yapılması tavsiye edilmektedir. Daha sonra, ECW formatına dönüştürülen görüntüler sunucuya kopyalanmalıdır. Ayrıca, sunucu ile kişisel bilgisayar arasındaki hız, sunucuya bağlanan kullanıcı sayısı ile ters orantılıdır. Örneğin, aynı anda görüntü kesme işlemi sunucu üzerinden birden fazla bilgisayardan başlatılırsa, Şekil 11'de verilen yüzdeler daha da artacaktır ve görüntü kesme süresi daha da artacaktır.

Şu andaki sorun, kesilen görüntülerin CAS sistemine yüklenmeden önce ECW formatına dönüştürülmesidir. ECW formatı Hexagon adlı ticari bir şirkete ait bir dosya formatıdır. Bu yüzden, görüntüleri bu formatta kaydetmek sadece bu şirketin geliştirdiği yazılımlar (ERDAS ve Global Mapper gibi) ile mümkündür. Bu yazılımlarda, ArcGIS yazılımındaki gibi bir programlama dili ile sorunsuz uyum bulunmamaktadır. Eğer ileride ECW formatında görüntü kaydetmek ArcGIS yazılımı ile mümkün olduğunda veya CAS sistemine ECW formatlı görüntü yükleme zorunluluğu değişirse, bu sorun ortadan kalkacaktır.

Bir diğer önemli konu ise, ArcGIS yazılımında, piramit seçeneği, açık olarak programcıklara önceden tanımlanmıştır ve genellikle bu seçenek kullanıcı tarafından değiştirilemez. Ancak, programın ortam seçeneklerini (environment settings) Python kod içinde elle değiştirmek mümkündür. Programcıkta piramit seçeneğini kapatılarak hem ortalama %60 civarı zaman kazanılmıştır hem de piramit dosyası oluşturulmadığı için bilgisayarda fazladan yer işgal etmemiştir.

Son konu ise Python dilinde geliştirilen kodlar genelde ArcGIS ArcToolbox'da bulunan programcılara göre çok daha hızlı çalışabilmektedir. Örneğin, Python da yazılan en büyük dikdörtgen bul kodu, görüntünün içinde kesilecek bölgenin 4 köşe koordinatlarını, 1 saniyenin altında bulabilmektedir. Bu kodlar, Linux tabanlı işletim sistemlerinde çalıştırılarak işlemler daha da hızlandırılabilir. Python dilinde görüntü işleme ve CBS işleri için birçok ücretsiz kütüphane bulunmaktadır ve bu kütüphanelerin Python ile uyumu Linux tabanlı işletim sistemlerinde, Windows tabanlı işletim sistemlerine göre çok güçlüdür. Bu kütüphaneler sayesinde görüntü kesme işlemi ArcGIS'teki programcılar kullanılmadan da yapılabilir. Böylelikle, görüntü ile işlemler, herhangi bir lisanslı yazılım kullanmadan çok hızlı bir biçimde gerçekleştirilebilir. Ayrıca, yazılan kodlar açık kaynaklı (open-source) olacağı için, bu kodların başka kullanıcılar tarafından da geliştirilme imkânı vardır.

#### KAYNAKLAR

Deng, M., Di L., Han W., Yagci A. L., Peng C., Heo G. (2013). **Web-service-based monitoring and analysis of global agricultural drought**, Photogrammetric Engineering & Remote Sensing, 79:10, 929-943. DOI: 10.14358/PERS.79.10.929.

Yagci, A. L., Santanello J. A., Jones J. W. (2017). **Estimating evaporative fraction from readily obtainable variables in magrove forests of the Everglades, U.S.A.**, International Journal of Remote Sensing, 38:14, 3981-4007. DOI: 10.1080/01431161.2017.1312033.

Yagci, A. L., Di L., Deng M., Yu G., Peng C., (2015). **Global agricultural drought mapping: results for the year 2011**, 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3764-3767. DOI: 10.1109/IGARSS.2012.6350498.

URL-1: <https://www.hgk.msb.gov.tr/kurumsal>  
Erişim Tarihi: 3 Ekim 2017.

URL-2: <https://www.hgk.msb.gov.tr/uf-11-uretim-is-akisi.html>  
Erişim Tarihi: 3 Ekim 2017

URL-3: <https://lpdaac.usgs.gov>  
Erişim Tarihi: 29 Kasım 2017

URL-4: <https://espa.cr.usgs.gov>  
Erişim Tarihi: 29 Kasım 2017

URL-5: <https://www.class.ncdc.noaa.gov>  
Erişim Tarihi: 29 Kasım 2017

URL-6: <http://desktop.arcgis.com/en/arcmap/10.3/tools/environments/compression.htm>  
Erişim Tarihi: 3 Ekim 2017