

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜMLEV İMGE TEMELLİ HIZLI EVRİŞİM İŞLEMLERİ

TOLGA TUNÇEL

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜMLEV İMGE TEMELLİ HIZLI EVRİŞİM İŞLEMLERİ

TOLGA TUNÇEL

Dr.Öğr. Üyesi Orhan AKBULUT
Danışman, Kocaeli Üniv.

.....

Prof.Dr. Yaşar BECERİKLİ
Jüri Üyesi, Kocaeli Üniv.

.....

Dr.Öğr. Üyesi Ali Can KARACA
Jüri Üyesi, Bandırma Onyediy Eylül
Üniv.

.....

Tezin Savunulduğu Tarih: 17.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasının amacı, görüntü işleme ve derin öğrenme alanlarında oldukça fazla kullanılan evrişim işlemlerinin, hesapsal karmaşıklığının tümlev imge gösterim yöntemi kullanarak düşürülmesidir.

Tez çalışmamda desteğini esirgemeyen, çalışmalarına yön veren, bana güvenen ve yüreklendiren danışmanım Dr. Öğr. Üyesi Orhan AKBULUT'a sonsuz teşekkürlerimi sunarım.

Hayatım boyunca beni her alanda destekleyen sevgili annem Nezahat TUNÇEL, babam Mustafa TUNÇEL ve kardeşim Süleyman TUNÇEL'e teşekkürlerimi sunarım.

Haziran – 2021

Tolga TUNÇEL

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	ii
SİMGELER VE KISALTMALAR DİZİNİ	vi
ÖZET.....	vii
ABSTRACT.....	viii
GİRİŞ	1
1. TÜMLEV İMGE GÖSTERİMİ	2
1.1. Tümlev İmge Nedir?	2
1.2. Tümlev İmgenin Oluşturulması.....	3
2. GÖRÜNTÜ İŞLEMEDE TÜMLEV İMGE KULLANIMI.....	11
2.1. Giriş	11
2.2. İlgili Çalışmalar	11
3. TÜMLEV İMGE TEMELLİ EVRİŞİM	17
3.1. Evrişimin Tanımı.....	17
3.2. Evrişim İşlemlerinin Kullanılması	19
3.3. Evrişim İşlemlerinin Tümlev İmge Kullanarak Gerçekleştirilmesi	22
4. DENEYSEL SONUÇLAR	29
5. SONUÇLAR	37
KAYNAKLAR	38
KİŞİSEL YAYINLAR VE ESERLER	41
ÖZGEÇMİŞ	42

ŞEKİLLER DİZİNİ

Şekil 1.1.	İmge içerisinde bulunan tümlev imge.....	2
Şekil 1.2.	Giriş imgesinin piksel görüntüsü	3
Şekil 1.3.	Tümlev imge oluşturmada birinci bölüm (a) Ham imge, (b) Birinci satır için tümlev imge.....	3
Şekil 1.4.	Tümlev imge oluşturmada ikinci bölüm (a) Ham imge, (b) Birinci sütun için tümlev imge	4
Şekil 1.5.	Tümlev imge oluşturma (a) Ham imge, (b) Ham imge için tümlev imge.....	4
Şekil 1.6.	Tümlev imge oluşturma-II (a) Ham imge, (b) Ham imge için tümlev imge.....	5
Şekil 1.7.	Ham imge için oluşturulan tümlev imge (a) Ham imge, (b) Ham imge için tümlev imge.....	5
Şekil 1.8.	Ham imgenin iç bölgesinde bulunan bir piksel değeri.....	6
Şekil 1.9.	Tümlev imge üzerinden kullanılacak üç piksel	6
Şekil 1.10.	Tümlev imge içerisinde bölge.....	7
Şekil 1.11.	Tümlev imge üzerinden toplama hesabında kullanılan sağ alt piksel.....	8
Şekil 1.12.	Tümlev imge üzerinden toplama hesabında kullanılan sol üst piksel	8
Şekil 1.13.	Tümlev imge üzerinden toplama hesabında kullanılan sağ üst piksel.....	9
Şekil 1.14.	Tümlev imge üzerinden toplama hesabında kullanılan sol alt piksel	9
Şekil 1.15.	Tümlev imgeyle hesaplama için örnek (a) Ham imge üzerinde bir alan, (b) Alan için tümlev imge çözümlemesinde kullanılacak piksel değerleri	10
Şekil 3.1.	Kenar algılama için çekirdek matrisi	17
Şekil 3.2.	Kenar algılama evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi.....	18
Şekil 3.3.	Keskinleştirme için kullanılan çekirdek matrisi	18
Şekil 3.4.	Keskinleştirme için evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi	18
Şekil 3.5.	Kutu bulanıklaştırma için çekirdek matrisi.....	19
Şekil 3.6.	Kutu bulanıklaştırma için evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi	19
Şekil 3.7.	Giriş imgesine eklenen katman (a) Giriş imgesi, (b) Çıkış imgesi.....	22
Şekil 3.8.	İlk piksel için evrişim	22
Şekil 3.9.	Gaussian bulanıklık süzgeci.....	24
Şekil 3.10.	Gaussian bulanıklık süzgecinin alt matrislere ayrıştırılması	24
Şekil 3.11.	Kenar tespit süzgecinin alt matrislere ayrıştırılması.....	25
Şekil 3.12.	Keskinleştirme süzgecinin alt matrislere ayrıştırılması	25
Şekil 3.13.	5×5 bulanıklık süzgecinin alt matrislere ayrıştırılması.....	26
Şekil 3.14.	Giriş imgesinin tümlev imgesi (a) Giriş imgesi, (b) Giriş imgesinin tümlev imgesi	26
Şekil 3.15.	Çekirdek matrisinin ilk parçasının tümlev imge ile çözümlenmesi.....	27
Şekil 3.16.	Çekirdek matrisinin ikinci parçasının tümlev imge ile çözümlenmesi.....	27

Şekil 3.17. Çekirdek matrisinin üçüncü parçasının tümlev imge ile çözümlenmesi.....	28
Şekil 4.1. 800×600 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması	31
Şekil 4.2. 1280×720 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması	32
Şekil 4.3. 1440×900 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması	33
Şekil 4.4. 2048×1080 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması	33
Şekil 4.5. 3840×2160 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması	34



TABLolar DİZİNİ

Tablo 4.1. Evriřimde kullanılan çekirdek matrisleri ve boyutları.....	29
Tablo 4.1. (Devam) Evriřimde kullanılan çekirdek matrisleri ve boyutları	30
Tablo 4.1. (Devam) Evriřimde kullanılan çekirdek matrisleri ve boyutları	31
Tablo 4.2. Farklı sayıda alt çekirdek matrisleri için evriřim süreleri.....	35



SİMGELER VE KISALTMALAR DİZİNİ

ms : Milisaniye
GB : Gigabyte



TÜMLEV İMGE TEMELLİ HIZLI EVRİŞİM İŞLEMLERİ

ÖZET

Evrışim işlemlerinde giriş görüntülerinin çözünürlükleri ve çekirdek matrislerinin boyutları hesapsal yük çözümlerinde önem arz eden unsurlar olduğu bilinmektedir. Giriş görüntülerinin çözünürlükleri veya çekirdek matrisinin boyutlarının artması, hesapsal yükün artmasıyla doğru orantılıdır. Bu durumlarda geleneksel evrişim yönteminin performansının yetersiz kaldığı gözlemlenmiştir. Bu nedenle, çekirdek matrislerinin boyutları veya giriş imgesinin çözünürlüğünün artması durumunda performans kaybı yaşamadan evrişim yapabilmek bu tez kapsamında geliştirilmiştir.

Evrışim işlemlerinde kullanılan çekirdek matrisleri doğrudan tümlev imge gösterim yöntemiyle kullanılmaya uygun olmayabilir. Tümlev imge gösterim yöntemiyle kullanıma uygun olmayan çekirdek matrislerini, alt matrislerin doğrusal birleşimi olarak yazmak mümkündür. Ayırıştırılan alt matrisler üzerinden tümlev imge gösterim yöntemiyle evrişim yapılabilmektedir.

Tez kapsamında, hesapsal karmaşıklığı değerlendirmek için 5 farklı çözünürlüğe sahip giriş görüntüsü ve 8 farklı boyutta çekirdek matris belirlenmiştir. Giriş görüntüleri üzerinden, belirlenmiş olan 8 farklı çekirdek sırasıyla geleneksel evrişim ve önerilen yöntemle hesaplanmış ve hesaplama süreleri karşılaştırılmıştır. Geleneksel evrişim işlemlerinin hesapsal karmaşıklığı çekirdek matrisinin boyutuyla doğru orantılı olarak artmaktadır. Tümlev imge temelli evrişim işlem süresinin çekirdek matris boyutuyla önemli bir ilişkisi olmadığı görülmektedir. Evrişimde kullanılan çekirdek matrislerinin boyutları arttıkça tümlev imge temelli evrişimin geleneksel evrişime göre daha az hesapsal karmaşıklığa sahip olduğu görülmüştür.

Anahtar Kelimeler: Çekirdek Matrisleri, Evrişim, Tümlev İmge.

INTEGRAL IMAGE BASED FAST CONVOLUTION OPERATIONS

ABSTRACT

It has known that the resolution of the input images and the dimensions of the kernel matrices are significant factors in computational load analysis in convolution operations. Increasing the resolution of the input images or the size of the kernel matrix is directly proportional to the increase in the computational load. In these cases, the performance of the traditional convolution method is insufficient. For this reason, it has been developed within the scope of this thesis to be able to convolution without losing performance in case the size of kernel matrices or the resolution of the input image increase.

Kernel matrices used in convolution operations may not be appropriate for direct use with the integral image method. Kernel matrices that are not inappropriate for use with the integral image method as linear combinations of sub-matrices. Convolutions can be made over the separated sub-matrices with the integral image method.

In this thesis, computational complexity have evaluated with 5 different resolution input image and 8 different kernel matrix. The input images were calculated with the traditional convolution and the proposed method, with 8 different kernels determined, and the calculation times had compared. The computational complexity of traditional convolution operations increases with the size of kernel matrix. It has seen that the integral image based convolution processing time has no significant relationship with the kernel matrix size. As the size of kernel matrices increases, it has seen that the integral image based convolution has less computational complexity than the traditional convolution.

Keywords: Kernel Matrices, Convolution, Integral Image.

GİRİŞ

Günümüzde dijital teknolojinin düzenli olarak gelişmesi, birçok alanda yenilikleri beraberinde getirmiştir. Bu alanlardan biri olan görüntü işleme, günümüzde robotik sistemler, tıp'ın farklı alanları, biyomedikal, savunma sanayi, coğrafi bölge bilgileri, imalat gibi hayatın birçok alanında kullanılmaktadır [1]. Görüntü işleme, dijital ortama aktarılarak görseller üzerinden ihtiyaca göre yararlı bilgiler çıkarmak için kullanılan yöntemdir. Aygıtlardan alınan veriler bir sunucuda toplanarak işlenebilmektedir. Veri miktarının artması ve bu verilerin işlenmesi, hesapsal yükün artmasını beraberinde getirmektedir. Bundan dolayı, hesapsal yükü azaltmaya yönelik yöntemler önem kazanmıştır.

Bu yöntemlerden biri olan tümlev imge gösterim yöntemi, bir imge içerisinde karesel veya dikdörtgensel bir bölgenin altında kalan piksellerin hızlı bir şekilde toplanmasını mümkün kılarak hesapsal yükü azaltan bir yaklaşımdır. Tümlev imge gösterimi literatürde farklı alanlarda kullanılmaktadır.

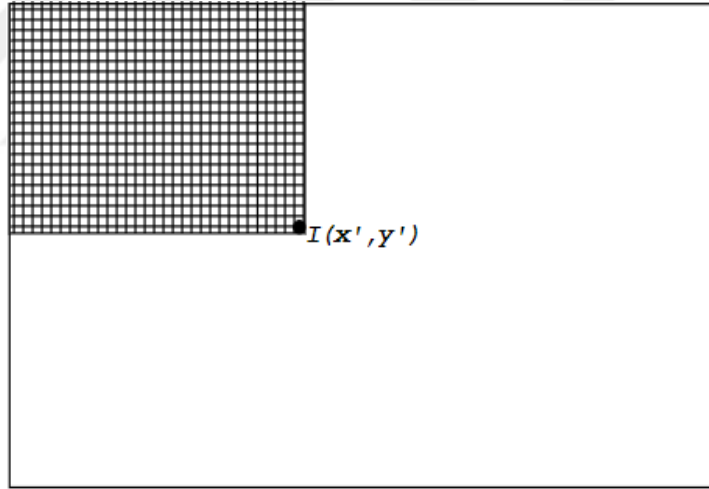
Bu tez çalışmasında, görüntü işlemede yaygın olarak kullanılan evrişim işlemlerinin hesapsal karmaşıklığını azaltmak için tümlev imge yönteminin kullanılması önerilmiştir. Normal ve yüksek çözünürlüğe sahip görüntüler üzerinden farklı çekirdek matrisiyle, geleneksel evrişim ve tez kapsamında önerilen tümlev imge tabanlı evrişim yöntemlerinin performansı karşılaştırılması yapılmıştır. Performans analizi olarak evrişim işlemlerinin hesapsal karmaşıklığı baz alınmıştır.

Tezin 1. bölümünde tümlev imge kavramından bahsedilmiş ve tümlev imgenin nasıl elde edildiği anlatılmıştır. 2. bölümde tümlev imge gösteriminden faydalanılarak yapılan akademik çalışmalar derlenmiştir. 3. bölümde evrişim özetlenmiş, literatürde ki çalışmalardan bahsedilmiş ve önerilen yöntem sunulmuştur. 4. bölümde geleneksel evrişim ve tümlev imge temelli evrişim yöntemlerinin hesapsal karmaşıklık sonuçları verilmiştir. Son bölümde ise sonuçlara yer verilmiştir.

1. TÜMLEV İMGE GÖSTERİMİ

1.1. Tümlev İmge Nedir?

Görüntü işleme uygulamalarında blok temelli işlemler yaygın bir şekilde kullanılmaktadır. Bu işlemlerin başında, blok üzerinden yapılan toplama işlemleri gelmektedir. Toplama veya ortalama alma işlemlerinin hesapsal yükü blok boyutuna bağlı olarak değişmektedir. Tümlev imge gösterim yaklaşımı, blok boyutundan bağımsız toplama veya ortalama alma işlemlerini gerçekleştirmeyi mümkün kılmaktadır. Bu yaklaşımın ana görevi, bir imge içinde karesel veya dikdörtgensel bir bölgenin içinde kalan alanın toplanmasını hızlı bir şekilde yapmaktır. Bir imge için I bütünleşik tümlev imge resmi olarak tanımlanır.



Şekil 1.1. İmge içerisinde bulunan tümlev imge

Şekil 1.1’de gösterilen dikdörtgensel bölge içinde kalmış piksel değerlerin toplamı tümlev imgede (x',y') noktasında ki değere eşittir.

Tümlev imge gösteriminin hesaplanması Eşitlik (1.1)’de verilmiştir;

$$I(x',y') = \sum_{x < x', y < y'} I(x,y) \quad (1.1)$$

Eşitlik (1.1)’de gösterilen $I(x,y)$ ham imgeyi gösterirken, $I(x',y')$ tümlev imgeyi göstermektedir.

1.2. Tümlev İmgenin Oluşturulması

Giriş imgesi içerisinde bulunan bir alanın tümlev imgeyle hesabının yapılabilmesi için giriş imgesinin ilgili alandaki tüm piksel değerlerinin kullanılması gerekmektedir. Şekil 1.2’de bir giriş imgesinin piksel görüntüsü gösterilmektedir.

4	3	7	8	3	2
3	6	4	5	7	8
5	6	4	7	2	5
5	7	4	2	6	8
0	3	2	4	6	8
1	4	9	0	5	7

Şekil 1.2. Giriş imgesinin piksel görüntüsü

Giriş imgesinin birinci satırın ilk elemanı veya birinci sütunun ilk elemanı tümlev imgenin ilk elemanı olarak yazılmaktadır. Birinci satırın, ilk elemanı ve ikinci elemanı toplama işlemi yapılarak tümlev imgenin birinci satırının ikinci elemanı olarak yazılmaktadır. Birinci satır tamamlanana kadar bu işlem devam etmektedir. Şekil 1.3’te tümlev imgenin birinci bölümünün oluşturulması gösterilmektedir.

4	3	7	8	3	2	4	7	14	22	25	27
3	6	4	5	7	8						
5	6	4	7	2	5						
5	7	4	2	6	8						
0	3	2	4	6	8						
1	4	9	0	5	7						

(a)

(b)

Şekil 1.3. Tümlev imge oluşturmada birinci bölüm (a) Ham imge, (b) Birinci satır için tümlev imge

Daha sonra birinci sütunun, ilk elemanı tümlev imge oluşturmanın birinci bölümünde olduğu gibi sabit kalmaktadır. Birinci sütunun, ilk elemanı ile ikinci elemanı toplama işlemi yapılarak imgenin birinci sütunun ikinci elemanı olarak yazılmaktadır. Birinci sütun tamamlanana kadar bu işlem devam etmektedir. Sonuç olarak birinci satır için yapılan işlem, birinci sütun içinde yapılmaktadır. Şekil 1.4'te tümlev imgenin ikinci bölümünün oluşturulması gösterilmektedir.

4	3	7	8	3	2	4	7	14	22	25	27
3	6	4	5	7	8	7					
5	6	4	7	2	5	12					
5	7	4	2	6	8	17					
0	3	2	4	6	8	17					
1	4	9	0	5	7	18					

(a) (b)

Şekil 1.4. Tümlev imge oluşturmada ikinci bölüm (a) Ham imge, (b) Birinci sütun için tümlev imge

Birinci satır ve birinci sütun işlemleri tamamlandıktan sonra iç bölgede kalan piksel değerleri 2 farklı yöntem ile hesaplanabilmektedir. İç bölgede kalan piksel değerleri için uygulanan birinci yöntemde, ham imge üzerinden birinci satırın ilk iki elemanı ve ikinci satırın ilk iki elemanı veya birinci sütunun ilk iki elemanı ve ikinci sütunun ilk iki elemanı toplanarak elde edilen piksel değeri, tümlev imgenin ikinci satır ve ikinci sütunda bulunan piksel değerine yazılır. Şekil 1.5'te bu işlem gösterilmektedir.

4	3	7	8	3	2	4	7	14	22	25	27
3	6	4	5	7	8	7	16				
5	6	4	7	2	5	12					
5	7	4	2	6	8	17					
0	3	2	4	6	8	17					
1	4	9	0	5	7	18					

(a) (b)

Şekil 1.5. Tümlev imge oluşturma (a) Ham imge, (b) Ham imge için tümlev imge

Benzer bir hesaplama Şekil 1.6’da gösterilmiştir. Ham imge üzerinden birinci satırın ilk üç elemanı ve ikinci satırın ilk üç elemanı veya birinci sütunun ilk iki elemanı, ikinci sütunun ilk iki elemanı ve üçüncü sütunun ilk iki elemanı toplanarak tümlev imgenin ikinci satır ve üçüncü sütununda bulunan piksel değerine yazılır.

4	3	7	8	3	2
3	6	4	5	7	8
5	6	4	7	2	5
5	7	4	2	6	8
0	3	2	4	6	8
1	4	9	0	5	7

(a)

4	7	14	22	25	27
7	16	27			
12					
17					
17					
18					

(b)

Şekil 1.6. Tümlev imge oluşturma-II (a) Ham imge, (b) Ham imge için tümlev imge

Bu işlemler ham imge üzerinde bulunan tüm piksel değerleri için uygulanarak tümlev imge oluşturulmaktadır. Şekil 1.7’de ham imge için oluşturulmuş tümlev imge gösterilmektedir.

4	3	7	8	3	2
3	6	4	5	7	8
5	6	4	7	2	5
5	7	4	2	6	8
0	3	2	4	6	8
1	4	9	0	5	7

(a)

4	7	14	22	25	27
7	16	27	40	50	60
12	27	42	62	74	89
17	39	58	80	98	121
17	42	63	89	113	144
18	47	77	103	132	170

(b)

Şekil 1.7. Ham imge için oluşturulan tümlev imge (a) Ham imge, (b) Ham imge için tümlev imge

Tümlev imge oluşturmada kullanılan ikinci yöntem, ilk yöntemle göre daha hızlı ve pratiktir. Bu yöntemde, temel olarak oluşturulan tümlev imge görüntüsü üzerinden üç piksel değeri ve ham imgeden bir piksel değeri kullanılmaktadır. Uygulanan iki

yöntemde de Şekil 1.3 ve Şekil 1.4'te gösterilen adımlar ortaktır. Şekil 1.8'de ham imgenin iç bölgesinde bulunan bir piksel değeri gösterilmiştir.

4	3	7	8	3	2
3	6	4	5	7	8
5	6	4	7	2	5
5	7	4	2	6	8
0	3	2	4	6	8
1	4	9	0	5	7

Şekil 1.8. Ham imgenin iç bölgesinde bulunan bir piksel değeri

Şekil 1.8'de mavi renkle gösterilmiş olan piksel değerinin bulunduğu noktanın tümlev imgesinin hesaplanması için ham imgenin o noktada ki piksel değeri kullanılmaktadır. Oluşturulacak tümlev imge üzerinden kullanılacak üç piksel değeri Şekil 1.9'da gösterilmiştir.

Oluşturulması istenen piksel değerinin, üzerinde, solunda ve ayna simetriğinde bulunan pikseller kullanılacaktır. Bu değerler Şekil 1.9'da bordo, mavi ve turuncu olarak gösterilmektedir.

4	7	14	22	25	27
7	16	27	40	50	60
12	27	42	62	74	89
17	39	58	80		
17	42	63			
18	47	77			

Şekil 1.9. Tümlev imge üzerinden kullanılacak üç piksel

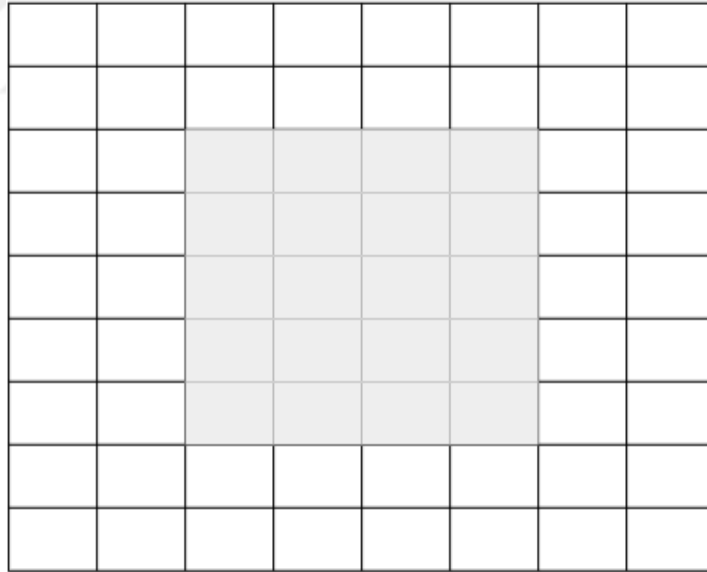
İç bölgede bulunan piksel değeri için tümlev imge, Şekil 1.8’de gösterilen ham imge üzerinde bulunan mavi renkli piksel değeri ve Şekil 1.9’da gösterilen tümlev imge üzerinde bulunan üç piksel değeri ile dört işlemde hesaplanabilmektedir. Şekil 1.8’de ham imge üzerinde bulunan mavi renkle gösterilen piksel değeri, Şekil 1.9’da bulunan bordo ve mavi renkli piksel değerleriyle toplanarak turuncu renkli piksel değerinden çıkartılmasıyla elde edilmektedir. Hesaplama işlemi Eşitlik (1.2)’de gösterilmiştir;

$$2+62+58-42=80 \quad (1.2)$$

Tümlev imgenin bir kere oluşturulduğunun bilinmesi önemlidir. Tümlev imge oluşturulduktan sonra hesaplanması istenen bir bölgenin alanı 4 işlemde hesaplanabilmektedir. Hesaplama işlemi Eşitlik (1.3)’te gösterilmiştir;

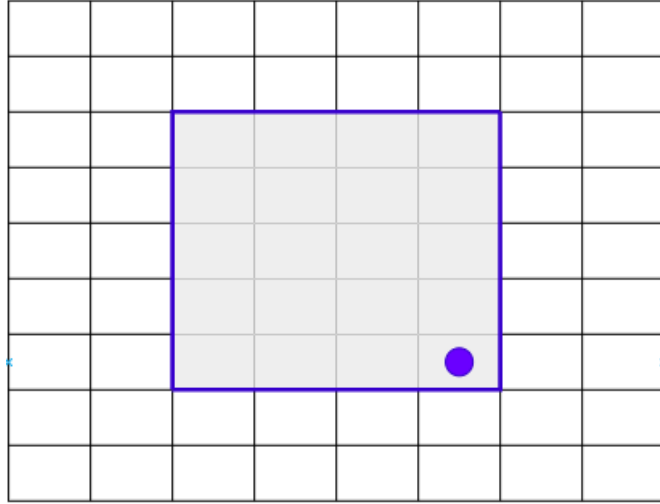
$$\sum_{(x,y) \in D} I(x,y) = I^f(x_2,y_2) + I^f(x_1,y_1) - (I^f(x_1,y_2) + I^f(x_2,y_1)) \quad (1.3)$$

Şekil 1.10’da tümlev imge içerisinde bir bölge gösterilmiştir.



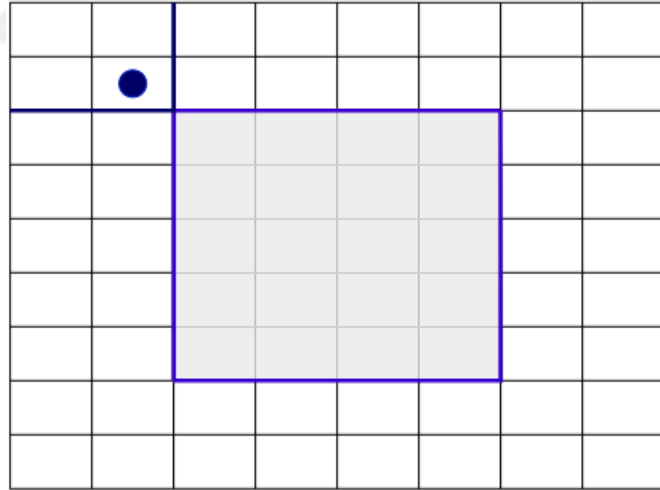
Şekil 1.10. Tümlev imge içerisinde bölge

Şekil 1.10’da gri renkle gösterilen bölgenin altında kalan piksel değerlerinin toplama hesabında kullanılacak ilk piksel değeri, gri alanın sağ altında bulunan piksel değerine denk gelmektedir. Bu piksel Şekil 1.11’de mor nokta olarak gösterilmiştir.



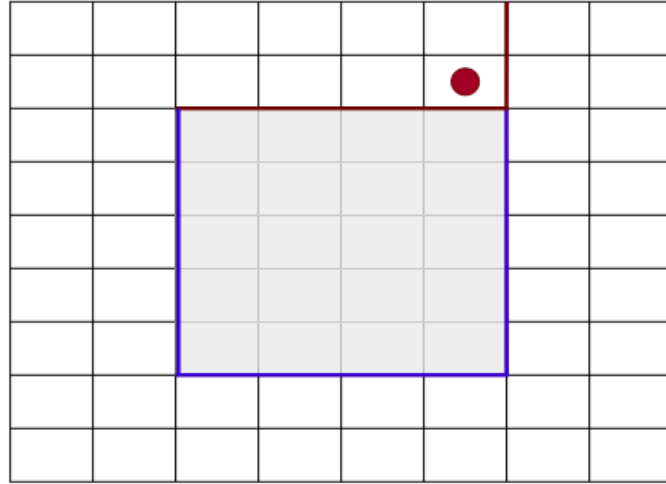
Şekil 1.11. Tümlev imge üzerinden toplama hesabında kullanılan sağ alt piksel

Tümlev imge üzerinden toplama hesabında kullanılacak bir diğer piksel değeri, Şekil 1.10'da gösterilen gri bölgenin sol üstünde bulunan piksel değerinin aynaya göre simetriğinde bulunan piksel değeridir. Bu piksel değeri Şekil 1.12'de lacivert nokta olarak gösterilmiştir.



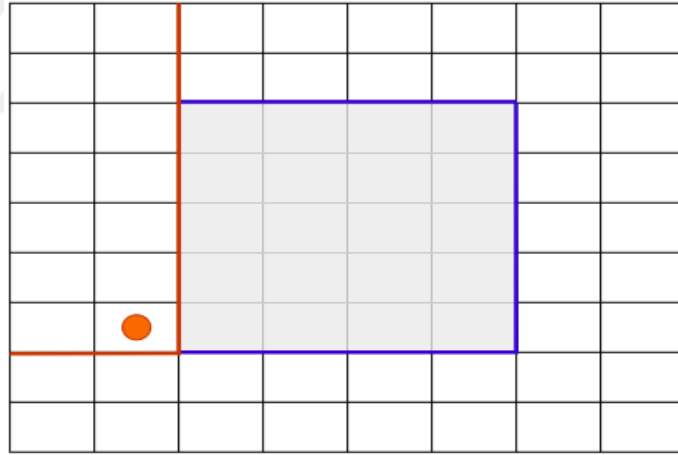
Şekil 1.12. Tümlev imge üzerinden toplama hesabında kullanılan sol üst piksel

Daha sonra Şekil 1.11'de mor noktayla gösterilen piksel değerinden lacivert noktayla gösterilen piksel değeri toplanmaktadır. Tümlev imge üzerinden toplam hesabında kullanılacak üçüncü piksel değeri, Şekil 1.10'da gösterilen gri bölgenin sağ üst piksel değerinin bir üstünde bulunan piksel değeridir. Bu piksel değeri Şekil 1.13'te bordo nokta olarak gösterilmektedir.



Şekil 1.13. Tümlev imge üzerinden toplama hesabında kullanılan sağ üst piksel

Tümlev imge üzerinden toplama hesabında kullanılacak son piksel değeri, Şekil 1.10'da gösterilen gri bölgenin sol alt piksel değerinin bir solunda bulunan piksel değeridir. Bu piksel değeri Şekil 1.14'te turuncu nokta olarak gösterilmektedir.



Şekil 1.14. Tümlev imge üzerinden toplama hesabında kullanılan sol alt piksel

Daha sonra, turuncu noktada bulunan piksel değeri Şekil 1.13'te gösterilen bordo noktada bulunan piksel değeriyle toplanarak mor ve lacivert noktada bulunan piksel değerlerinin toplamlarından çıkartılarak sonuca ulaşılmaktadır.

Şekil 1.10'da gösterilen gri bölgenin alanı geleneksel toplama yöntemiyle 20 adımda hesaplanabilirken aynı bölgenin alanı, tümlev imgeyle 4 adımda hesaplanabilmektedir. Şekil 1.15'te ham imge üzerinde bir alan ve gösterilen alanın tümlev imgeyle çözümlenmesinde kullanılacak piksel değerleri gösterilmiştir.

4	3	7	8	3	2	4	7	14	22	25	27
3	6	4	5	7	8	7	16	27	40	50	60
5	6	4	7	2	5	12	27	42	62	74	89
5	7	4	2	6	8	17	39	58	80	98	121
0	3	2	4	6	8	17	42	63	89	113	144
1	4	9	0	5	7	18	47	77	103	132	170

(a)

(b)

Şekil 1.15. Tümlev imgeyle hesaplama için örnek (a) Ham imge üzerinde bir alan, (b) Alan için tümlev imge çözümlemesinde kullanılacak piksel değerleri

Tümlev imgeyle Şekil 1.15'te gösterilen alanın hesabı Eşitlik (1.4)'te gösterilmiştir;

$$(113+16)-(50+42)=37 \quad (1.4)$$

2. GÖRÜNTÜ İŞLEMEDE TÜMLEV İMGE KULLANIMI

2.1. Giriş

Blok boyutlarından bağımsız olarak bir blok üzerinden hızlı bir şekilde toplama işlemi yapılabilmesi, hesapsal yükü önemli derecede azaltmaktadır. Bu yüzden blok tabanlı görüntü işleme uygulamalarında tümlev imgenin kullanımı oldukça yaygındır.

Bölüm 2.2’de tümlev imge gösteriminin kullanıldığı çalışmalar anlatılmıştır.

2.2. İlgili Çalışmalar

[2]’de imgeler üzerinden son derece hızlı ve yüksek oranlarda doğruluğa sahip algılama yapabilen yüz tanıma yöntemi geliştirilmiştir. Çalışma içerisinde eğitim için kullanılan tüm örnek alt pencerelere aydınlatma koşullarının etkisini azaltmak için varyans normalleştirilmiştir. Alt örnek pencerelerinin ortalamalarının hesabı için tümlev imge kullanılmıştır.

[3]’te kartezyen veri uzayı içerisinde bulunan muhtemel tüm hedef bölgelerin histogramlarını hesaplayabilmek için integral histogram adını verdikleri yeni bir yöntem sunulmuştur. Önerilen integral histogram yöntemi, histogram hesabının tümlev imgeyle hesaplanarak kullanışlı olmayan gerçek zamanlı olarak kapsamlı bir arama sürecini kullanmayı mümkün kılmaktadır.

Bradley ve arkadaşları [4] tümlev imge kullanarak gerçek zamanlı uyarlanabilir eşikleme için yeni bir teknik sunmuşlardır. Giriş görüntülerinin hesabı tümlev imgeyle hesaplanmaktadır. Önerilen yöntem aydınlatma farklılıklarına rağmen doğru eşikleme yapabilmektedir.

[5]’te imgeleri son derece hızlı ve yüksek oranlarda sezme kapasitesine sahip, nesne algılama tabanlı makine öğrenmesi yaklaşımı sunulmuştur. Önerilen yöntem, karmaşık sınıflandırıcıları bir kademe içerisinde birleştirmeye olanak sağlamaktadır.

Bu sayede imge içerisinde bulunan arka planların hızlı bir şekilde atılabilmektedir. Çalışma içerisindeki detektör tarafından kullanılan özelliklerin hızlı bir şekilde hesaplanması tümlev imge gösterim yöntemiyle yapılmıştır.

[6]'da nesne algılama ve doku sınıflandırması üzerinde kullanılan yeni bir bölge tanımlayıcısı oluşturulmuştur. Nesne algılama ve doku sınıflandırması için bölgesel ortak değişinti matrisi tanımlanmıştır. Tanımlanan matris doğrudan tümlev imge yöntemiyle hesaplanamamaktadır. Bu sorunu aşmak için Eşitlik (2.1)'de gösterilen geleneksel kovaryans matrisini;

$$C_R(i,j) = \frac{1}{n-1} \sum_{k=1}^n (z_k(i) - \mu_k(i)) (z_k(j) - \mu_k(j)) \quad (2.1)$$

bazı matematiksel işlemlerden geçirek Eşitlik (2.2)'de gösterilen duruma getirerek;

$$C_R(i,j) = \frac{1}{n-1} \left[\sum_{k=1}^n z_k(i)z_k(j) - \frac{1}{n} \sum_{k=1}^n z_k(i) \sum_{k=1}^n z_k(j) \right] \quad (2.2)$$

tümlev imgeyle hesaplanabileceği gösterilmiştir. Giriş imgesi üzerinden tümlev imge oluşturulabildiği gibi imge üzerinden çıkarılan öznelikler içinde tümlev imge elde edilmiştir.

Jiang ve arkadaşları [7] son yıllarda hesaplama hızının önemli hale geldiği DIC (Digital Image Correlation) algoritmasının tümlev imge gösterim yöntemiyle hesapsal karmaşıklığını düşürerek hesaplama sürecinin hızlandırılması üzerine çalışmışlardır. Enterpolasyon yönteminin DIC algoritması üzerindeki etkisi de araştırılmıştır. Sonuçlar incelendiğinde, tümlev imge uygulanan DIC algoritmasının hesaplama hızını büyük ölçüde arttırdığı ve hızlı enterpolasyon yöntemi kullanıldığında iyileşmelerin daha net olduğu gözlemlenmiştir.

[8]'de yüksek alan ve güç verimliliği ile hızlandırılmış tümlev imge hesaplaması için bir yöntem önerilmiştir. Tümlev imgenin hesaplanmasındaki veri akışı performansını artırmak için çift yönlü veri odaklı hesaplama işleyişi önerilmiş ve bu işleyişi destekleyecek borulu paralel mimari tasarlanmıştır. Geliştirilen mimari, güç tüketimi ve düşük kaynak kullanımıyla işlem hızının oldukça yüksek olduğu görülmüştür.

[9]'da geleneksel yöntemlerin çok yönlü görüntülerde kişi algılama işlemlerinde başarısız olduğu belirtilmiştir. Başarısızlığı ortadan kaldırmak için entegre kanal özelliklerine dayanan bir yöntem geliştirilmiştir. LUV ve gradyan büyüklüğü gibi

farklı kanallardan entegre kanal özellikleri çıkartılmıştır ve güçlendirilmiş karar ağaçları kullanılarak sınıflandırılmışlardır. Halka dilimi şeklinde bulunan dairesel sektörlerin içerisinde bulunan piksel toplamları özellikleri ifade etmektedir. Dairesel sektörlerin içerisinde bulunan toplamları efektif bir şekilde hesaplamaya yarayan açısız tümlev imge yöntemi geliştirilmiştir.

Holzer ve arkadaşları [10] düzenli nokta bulutu verilerinden yüzey normallerini kestirmek için gerçek zamanlı çalışan yöntemler sunmuşlardır. Çalışma içerisinde yumuşatma ve kovaryans tahmini için tümlev imge yöntemi kullanılmıştır. Büyük nokta kümelerinden hızlı bir şekilde yüzey normalleri çıkarılmasını mümkün kıldığı için Kinect benzeri verilerle çalışan gerçek zamanlı görme algoritmalarında kullanılabileceği gösterilmiştir.

[11]'de hızlı ve güvenilir bir insan tanıma sistemi oluşturma hedeflenmiştir. Sistemde kullanılan kademeli rejektör yaklaşımını yönlendirilmiş gradyanların histogramları özellikleriyle entegre edilmiştir. AdaBoost ile geniş bir blok setinden uygun bloklar seçilmiştir. Yönlendirilmiş gradyan histogramlarının özneliklerinin hesaplama hızını, dikkate değer bir şekilde arttıran tümlev imge gösterim yöntemi kullanılmıştır. Oluşturulan sistemin doğruluk oranı mevcut sistemler ile yakınken, görüntü işleme hızının mevcut sistemlere göre daha yüksek olduğu belirtilmektedir.

[12]'de standart kabul edilen SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), FAST (Features from Accelerated Segment Test) ve Harris detektörleri gibi diğer detektörlere göre daha iyi performansa sahip olan, alan ölçekli diğer detektörlere göre daha iyi hesapsal karakteristik özelliklere sahip değişmez ölçekli merkez çevre detektörü sunulmaktadır. Çalışma içerisinde yamuk alanları iki farklı tümlev imge kullanarak oluşturulmuştur. Yöntemin gerçek zamanlı uygulanabildiği söylenmiştir.

[13]'te görüntü sensörleri için standart Haar benzeri özelliklerinden farklı olarak genişletilmiş Haar benzeri özellikleri ve 45 derece dönderilmiş Haar benzeri özellikler tanımlanmıştır. Döndürelen Haar benzeri özellikler tümlev imgeyle hesaplanmıştır. Tümlev imge hesaplamaları GPU (Graphics Processing Unit)'ya aktırılabilir. GPU üzerinde tümlev imge hesaplamaları CPU (Central Processing Unit) üzerinden

yapılan tümlev imge hesaplamalarına göre daha hızlı olduğu ve karmaşık sınıflandırıcıların kullanımına izin verdiği gösterilmektedir.

[14]'te doku analizi üzerine verimli yeni bir yöntem geliştirilmiştir. Geliştirilen yöntemde doku analizinde bulunan piksel değerleri tümlev imgeyle hesaplanmıştır. Çalışmada tümlev imge toplam tablo olarak isimlendirilmiştir.

Bay ve arkadaşları [15] ölçeği ve rotasyonu değişmeyen yeni bir detektör ve tanımlayıcı sunulmuşlardır. Bazı açılardan mevcut detektörlerle benzer sonuçlar alınmış olmasına rağmen hesaplama hızının diğer detektörlere göre daha hızlı olduğu belirtilmiştir. Hızlı olması, görüntü evrişimlerinin tümlev imge kullanarak yapılmasına dayandırılmaktadır. Çalışmada nesne tanıma uygulaması üzerinden diğer yöntemlerle kıyaslama yapılmıştır ve önerilen yöntemin tanıma oranının daha yüksek çıktığı gösterilmiştir.

[16]'da video duman algılama sistemlerinin performansını arttırmaya yönelik tümlev imge tabanlı kümülatif hareket modeli geliştirilmiştir. Sistem gerçek zamanlı çalıştığı için tahminleme yapmanın zorluğuna değinilmiştir. Bu nedenle sadece duman için hareket yönü tahminlemesi yapılmış ve hesapsal yükü azaltmak için yinelemeli işlem yerine bütünlük tümlev imge kullanılmıştır. Geliştirilen model tarafından tahmin edilen yönelim kusursuz olmamakla birlikte hızlı çalıştığı gösterilmiştir.

Jung ve arkadaşları [17] tümlev imgeyle hızlandırılmış şablon eşleme yöntemi önerilmişlerdir. Yöntem, geleneksel eşleme algoritmaları yerine bütünlük tümlev imge kullanmaktadır. Tümlev imgenin geleneksel yöntemlere göre düşük seviyede ki blok toplamları çok hızlı bir şekilde hesaplandığı, iyi olmayan adayların atılma hızının iki kat daha hızlı çalıştığı ve daha az belleğe ihtiyaç duyduğu belirtilmiştir. Önerilen yöntemin geleneksel yöntemlere göre daha hızlı çalıştığı gösterilmiştir.

[18]'de tüm görüntü bölgeleri için eşitsizlik tahmini yapan alan tabanlı stereo eşleştirme algoritması önerilmiştir. Stereo eşleştirmelerinde, destek penceresinin tahmini ve pencere şeklini veya piksel destek ağırlıklarını alt sahnelere uyarlamının zorluğundan bahsedilmiştir. Önerilen yöntem bu zorluğun çözümlenmesine katkı sağlamaktadır. Rastgele bir 2 boyutlu bölgede gerçekleştirilen eşleştirme maliyetini azaltmak için basit entegrasyon üzerinde hızlanma sağlayan ortogonal tümlev imge tekniği geliştirilmiştir.

Veksler ve arkadaşları [19] hızlı ve doğruluk oranı yüksek bir değişken pencere algoritması geliştirmişlerdir. Algoritma, pencere boyutlarının süreklilik özelliklerinden faydalanarak, pencere için yararlı bir pencere şekli ve boyutları bulmaktadır. Ayrıca tüm piksellerin tahmini olarak aynı eşitsizlikte olup olmadığını kontrol eden yeni bir pencere maliyetini tanımlamaktadır. Pencere maliyetlerinin hesabı tümlev imge kullanarak yapılmıştır. Bu sayede, algoritmanın verimliliğinin düzenli ve iyi olduğu gösterilmektedir.

[20]'de tümlev imge yöntemi kullanılarak blok eşleştirme uygulaması geliştirilmiştir. Çalışma içerisinde tümlev imge yöntemi kapsamlı bir şekilde incelenmiştir. Daha sonra blok eşleştirme için tümlev imge kullanarak algoritma geliştirilmiştir.

[21]'de tümlev imge gösteriminin, akış işleme uygulamasında tasarımı ve değerlendirmesi üzerine araştırma yapılmıştır. Tümlev imge gösteriminin büyük giriş görüntüleri için işleme süresinin GPU üzerinde çalışan akış programlaması kullanarak dikkate değer bir şekilde azaldığı gözlenmiştir. Oluşturulacak donanımlara daha fazla akış işlemleri eklenerek diğer CPU'lara oranla daha az güç harcanarak gerçek zamanlı performans göstereceği belirtilmiştir.

Mikhisor ve arkadaşları [22] bilgisayarlı görme uygulamalarında doğrudan kullanılmayan balıkgözü görüntülerinin düzeltilmesi için üç farklı yöntem analiz etmişlerdir. Çalışma içerisinde kullanılan yerel projeksiyon için üretilen eğri kenarlı yamalar, geleneksel tümlev imge gösterimiyle hesaplanamadığı belirtilmiştir. Bu yüzden geleneksel tümlev imgede kullanılan köşe değerlerinden farklı olarak herhangi bir yerel iz düşüm yaması üzerinden tümlev imge hesaplanabileceği gösterilmiştir. Geleneksel tümlev imge yöntemine göre yavaş olmasına rağmen gerçek zamanlı uygulamalarda hızlı çalıştığı belirtilmiştir.

[23]'te Log-Gabor dönüşümü ve tümlev imge kullanarak göz yerelleştirme üzerine yöntem önerilmektedir. Göz ve kaş bölgelerini bulmak için Log-Gabor dönüşümü kullanılmakta ve dönüşüm sonuçları tümlev imgeyle hesaplanmaktadır. Daha sonra orijinal yüz görüntüleri içerisindeki kaba bölgeler tümlev imge yöntemiyle hesaplanmaktadır. Bu sayede göz yerelleştirme hassasiyeti korunurken, süreç basite indirgenip hesaplama süresinin azaldığı vurgulanmaktadır.

[24]'te el yazısı görüntülerinden kural çizgilerini kaldırmak için hızlı ve güçlü bir yöntem sunulmuştur. El yazısı görüntülerinde kullanılan yatay ve dikey projeksiyon profillerinin özellikleri tümlev imge yöntemiyle hızlı bir şekilde hesaplanmaktadır. Daha sonra SVM (Support Vector Machine) eğitiminde kullanılacak büyük çaplı bir veri setinden destek vektör noktalarının seçimi için bir yöntem önerilmiştir.

Huang ve arkadaşları [25] nesne tespitinde kullanılan yönlendirilmiş gradyan histogramını modifiye ederek hızlandırmayı hedeflenmişlerdir. Geleneksel yönlendirilmiş gradyan histogramı üzerinde iki değişiklik yapılmıştır. Birincisi, yönlendirilmiş gradyanları hesaplamak için arama tablosu kullanılmıştır. İkincisi, yönlendirilmiş gradyan histogram özelliklerinin hesaplanması için tümlev imge yöntemi kullanılmıştır. Önerilen yöntemler kullanılarak nesne tespitinin hızlı bir şekilde yapıldığı belirtilmiştir.

3. TÜMLEV İMGE TEMELLİ EVRİŞİM

3.1. Evrişimin Tanımı

Evrişim, giriş olarak verilen imgeleri bulanıklaştırma, yumuşatma, keskinleştirme, kenar algılama gibi görüntü işleme yöntemleri için sıklıkla kullanılmaktadır. Bu işlemleri yapabilmesi için görüntü işleme yöntemine uygun çekirdek matrisine ihtiyaç duymaktadır. Temel olarak, giriş imgesinin her bir pikselinin çekirdek matrisiyle hesaplanarak çıkış imgesini oluşturma işlemidir. Evrişim işlemi genel olarak Eşitlik (3.1)'de gösterilmiştir;

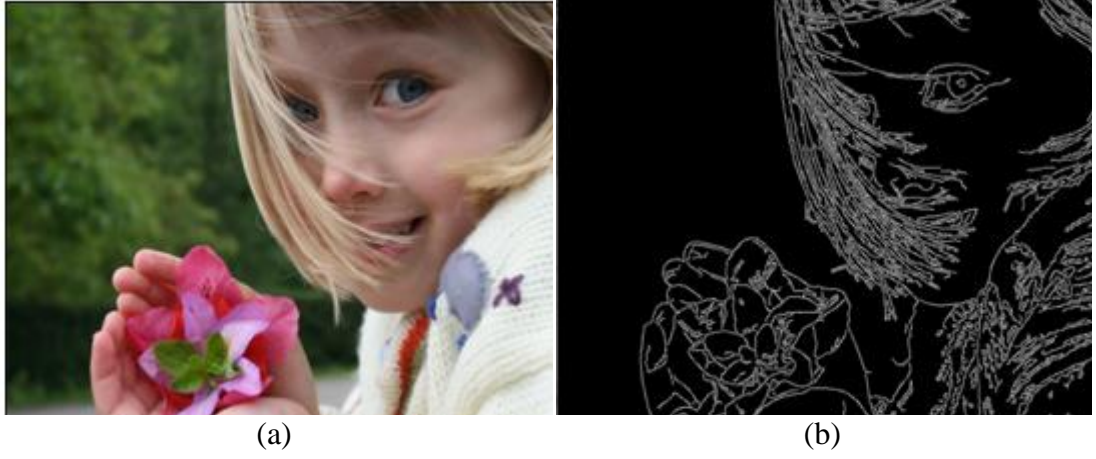
$$I_F(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(x-i,y-j)K(i,j) \quad (3.1)$$

I giriş görüntüsüne, K ise çekirdek matrisine karşılık gelmektedir. Çekirdek matrisi, süzgeçleme çekirdeği olarak da isimlendirilmektedir. Evrişim işlemlerinden farklı olarak çekirdek matrisleri, makine öğrenmesi alanında, imge üzerinden öznetelik çıkarımında kullanılmaktadır. Çekirdek matrisleri görüntü manipüle etme türüne göre farklılık göstermektedir. Şekil 3.1'de kenar algılama için kullanılan çekirdek matrisi gösterilmektedir. Giriş imgesi ve kenar algılama çekirdek matrisi evrişim yaparak çıkış imgesini oluşturmaktadır.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Şekil 3.1. Kenar algılama için çekirdek matrisi

Şekil 3.2'de kenar algılama için evrişim örneği gösterilmektedir.



Şekil 3.2. Kenar algılama evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi [26]

Şekil 3.3'te keskinleştirme için kullanılan çekirdek matrisi gösterilmektedir.

-1	-1	-1
-1	9	-1
-1	-1	-1

Şekil 3.3. Keskinleştirme için kullanılan çekirdek matrisi

Şekil 3.4'te keskinleştirme için evrişim örneği gösterilmiştir.



Şekil 3.4. Keskinleştirme için evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi [27]

Yaygın olarak kullanılan bir diğer çekirdek ise bulanıklaştırma evrişimlerinde kullanılan kutu bulanıklaştırma çekirdeğidir. Şekil 3.5'te kutu bulanıklaştırma çekirdeği gösterilmiştir.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Şekil 3.5. Kutu bulanıklaştırma için çekirdek matrisi

Şekil 3.6'da kutu bulanıklaştırma için evrişimin örneği gösterilmiştir.



(a)

(b)

Şekil 3.6. Kutu bulanıklaştırma için evrişim örneği (a) Giriş imgesi, (b) Çıkış imgesi

3.2. Evrişim İşlemlerinin Kullanılması

Girdi görüntüleri üzerinden anlamlı bilgileri çıkartmaya olanak tanıyan evrişim, literatürde farklı alanlarda kullanılmaktadır.

Keys ve arkadaşları [28] ayrık verileri örneklemek için kullanılan kübik evrişim enterpolasyonun yeni bir versiyonu geliştirmişlerdir. Çalışmada tek boyutlu enterpolasyon fonksiyonu oluşturulmuştur. Oluşturulan fonksiyon kullanılarak iki boyutlu enterpolasyonlar gerçekleştirilmiştir. En yakın komşu algoritması ve doğrusal

enterpolasyon yöntemine göre daha iyi sonuçlar verdiği söylenmiştir. Kübik eğrisi yaklaşımına göre hatalı sonuçlar verse de daha verimli çalıştığı gözlemlenmiştir.

[29]'da yoğun yıldızların küçük alt alanlarının türetilmesi için çalıştığı kanıtlanmış olan optimal çekirdek çözümlerini, değişken uzay çekirdekleri altında genelleştirilmiştir. Geliştirilen yöntem, farklı PSF (Point Spread Function) varyasyonları gibi yaygın olarak sorun çıkartan araçsal problemleri kendiliğinden düzelttiği görülmüştür.

Cabral ve arkadaşları [30] iki ve üç boyutlu vektör alanlarını görüntülemek için bir yöntem geliştirmişlerdir. Yöntem, bir vektör alanı boyunca doku filtrelemesi için kullanılabilir. Bir vektör alanı tarafından tanımlanan akış hatları boyunca giriş görüntüsünü filtreleyerek çıkış görüntüsü oluşturulmuştur. Teknik, farklı özel efektler oluşturabilmektedir. Basit ve verimli uygulamalara katkıda bulunabildiği söylenmiştir.

[31]'de evrişim tabanlı tıbbi enterpolasyonun geometrik dönüşümlere uygulanması ele alınmıştır. Çalışmada farklı enterpolasyon çekirdekleri farklı yöntemlere uygulanmıştır. Elde edilen döndürülmüş görüntülerde bulunan enterpolasyon hataları, referans görüntülerden kök-ortalama-karesi ve kök-tepe kare sapması yöntemleriyle analiz edilmiştir.

Chen ve arkadaşları [32] semantik görüntü segmentasyon uygulamasında filtrenin görüş açısının ayarlanması ve derin evrişim sinir ağları tarafından hesaplanan özellik dönüşlerinin çözünürlüklerini kontrol eden samsiyah evrişimi ele almışlardır. Çalışmada önerilen modül, eğitilmesi gereken toplu ve farklı hızlara sahip katmanlarla samsiyah evrişimden oluşmaktadır.

[33]'te renkli görüntülerin dörütlü-değerli maske ile birleştirilmesiyle yeni bir görüntü evrişimine dayanan, kenar algılama filtresi türü önerilmiştir. Fourier dönüşümleri kullanılarak renkli görüntü filtrelemesi üzerine yapılacak olan bir çalışmada renk dalgalanmasının gerektiği söylenmiş ve bu dalgalanma için çalışmada sunulan filtrenin tasarlandığı belirtilmiştir.

[34]'te eski derin sinir ağlarının, gelen özellikleri filtrelerle veya havuzlama sistemiyle çok ölçekli bilgileri kopyalamasını ve güncel derin sinir ağlarının, uzamsal bilgileri kademeli bir şekilde alarak nesne sınırlarını yakalayabilmesinin avantajlarını birleştiren bir yöntem önerilmiştir. Önerilen model nesne sınırları boyunca

segmentasyon sonuçlarını iyileştirmeyi hedeflemektedir. Ayrılabilir evrişimi, simsiyah uzamsal piramit havuzlama ve kod çözücü modüllere uygulayarak daha hızlı ve güçlü kodlayıcı-kod çözücü ağı tanımlanmıştır.

Sangwine ve arkadaşları [35] hiperkarmaşık maskelerle evrişim tabanlı doğrusal vektör filtresi sunmuşlardır. Prewitt, Sobel ve Kirsch filtrelerinden yola çıkarak renkli kenar algılama filtresi geliştirilmiştir. Çok bantlı görüntülere veya hiperkarmaşık katsayılı maske ile evrişim tabanlı filtrenin ilk örneklerinden olduğu belirtilmiştir.

Chen ve arkadaşları [36] öznitelik çıkarımı için örneklenmiş filtrelerle simsiyah evrişim uygulayarak semantik segmentasyona yönelik yenilenmiş bir ağı oluşturmayı hedeflemiştir. Yöntem, hesapsal yükü arttırmadan filtrelerin görüş alanını arttırabilmektedir. Nesnelerin birden fazla ölçekte bölümlere ayırabilmek için ASPP (Atrous Spatial Pyramid Pooling) kullanılmış ve birden çok görüntü bağlamı yakalanabilmiştir.

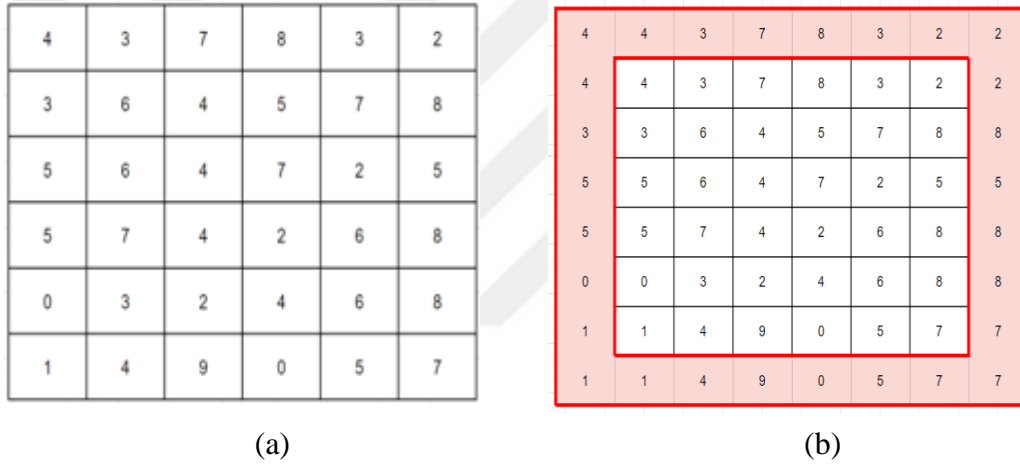
[37]'de vektör alanı evrişimi adı verilen aktif modeller için yeni bir dış kuvvet önerilmiştir. Sınırlı yakalama aralığı, gürültü hassasiyeti gibi performansı etkileyecek sorunlar ele alınmıştır. Vektör alan evrişim alanının, vektör alan çekirdeğinin görüntüden çıkarılan kenar haritalarıyla çevreleyerek hesaplanmakta olduğu belirtilmiştir.

[38]'de değişken kenar yapılarına uyum sağlayabilen yeni bir kübik evrişim enterpolasyon şeması önerilmiştir. Standart kübik evrişim enterpolasyonda ortaya çıkabilen bulanıklık, engelleme, çınlama gibi problemleri yeni şema ile çözümlenmesi hedeflenmiştir. Yöntem, hem keskin hem de doku ayrıtlarını koruyabilmektedir.

Fan ve arkadaşları [39] görüntü restorasyon performansını arttırabilmek için yeni bir yöntem geliştirmişlerdir. Ölçek değişmezliğinin, sinir ağlarına uygun bir şekilde modellendiği zaman görüntü restorasyon performansında önemli bir artış olduğu belirtilmiştir. Ölçek değişmezliği için farklı ölçeklerle evrişim yapmak üzere ölçek evrişimi önerilmiştir. Önerilen yöntem görüntü sıkıştırma ve görüntü çözünürlüğü gibi görüntü restorasyon işlemlerinde üstün performans sağlamaktadır.

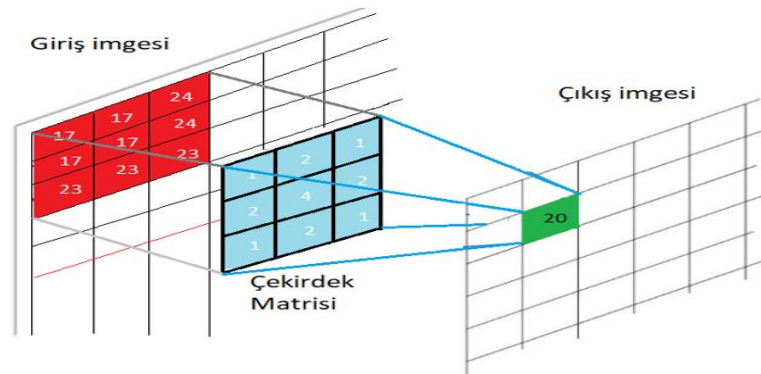
3.3. Evrişim İşlemlerinin Tümlev İmge Kullanarak Gerçekleştirilmesi

Evrişime giriş imgesinin ilk piksel değerinin çekirdek matrisiyle evrişimi sonucunda elde edilen değer, çıkış imgesinin ilk piksel değerine yazılarak başlanmaktadır. Bu işlem tüm giriş imgesinde bulunan pikseller için yapılarak çıkış imgesi oluşturulmaktadır. Fakat giriş imgesinin kenarlarında bulunan piksel değerlerine çekirdek matrisinin yapısı nedeniyle doğrudan evrişim uygulanamamaktadır. Bu sorunu çözebilmek için giriş imgesinin kenarlarında bulunan piksel değerleri değiştirilmeden katman olarak giriş imgesine eklenir. Evrişim için kullanılacak çekirdek matrisinin büyüklüğü ile oluşturulan katman sayısı doğru orantılıdır. Şekil 3.7’de kırmızı dikdörtgenler arasında kalan bölge, eklenen katmanı göstermektedir.



Şekil 3.7. Giriş imgesine eklenen katman (a) Giriş imgesi, (b) Çıkış imgesi

Giriş imgesine katman eklendikten sonra kenar değerleri için evrişim yapmak mümkündür. Şekil 3.8’de giriş imgesinin ilk piksel değerinin çekirdek matrisiyle evrişim modeli gösterilmektedir.



Şekil 3.8. İlk piksel için evrişim

Şekil 3.8’de gösterilen giriş imgesinin sol üst piksel değerinin evrişim hesabı Eşitlik (3.1)’den yararlanarak yapılmıştır. Bu hesaplama işlemi Eşitlik (3.2)’de gösterilmektedir;

$$[(23 \times 1) + (23 \times 2) + (23 \times 1) + (17 \times 1) + (17 \times 2) + (24 \times 1) + (17 \times 2) + (17 \times 4) + (24 \times 2)] \frac{1}{16} = 20 \quad (3.2)$$

Giriş görüntüsünde bulunan tüm pikseller için çekirdek matrisiyle evrişim yapılarak çıkış imgesi oluşturulmaktadır.

Tümlev imge yönteminin doğrudan evrişim işlemlerine uygulanması oldukça nadir görülen bir durumdur. Şekil 3.5’de imge bulanıklaştırma evrişiminde yaygın olarak kullanılan kutu bulanıklık süzgeciyle yapılan evrişim buna örnek verilebilir. Kutu bulanıklık süzgecinin tüm eleman değerlerinin 1 olmasından dolayı bu süzgeçle yapılan evrişim sonucunda elde edilen değer, geleneksel bölgesel ortalama alma işlemiyle de elde edebilmektedir. Geleneksel ortalama alma işleminin yapısı tümlev imge gösterimin yapısına uygun olduğu için doğrudan tümlev imge kullanarak kayıpsız ve hızlı bir şekilde hesaplanabilmektedir. Fakat evrişimde kullanılan birçok çekirdek matrisine doğrudan tümlev imge uygulanamamaktadır.

$G_{n \times n}$ 2-boyutlu bir çekirdek matrisi olarak tanımlansın. $l \in \{1, 2, 3, \dots, m\}$ olmak üzere $G_{n \times n}$ matrisi Eşitlik (3.3)’te olduğu gibi verilsin;

$$G_{n \times n} = G_{n \times n}^{l=1} + G_{n \times n}^{l=2} + \dots + G_{n \times n}^{l=m} \quad (3.3)$$

Evrişim işlemlerinin dağılma özelliğini kullanarak $G_{n \times n}$ çekirdek matrisini Eşitlik (3.3)’te gösterildiği gibi alt matrislere ayrıştırılabilmektedir. Ayrıştırılmış her bir $G_{n \times n}^{l=i}$ çekirdek matrisinin evrişimi, tümlev imge ile hesaplanabilirse, $G_{n \times n}$ çekirdek matrisinin evrişimi tümlev imge üzerinden hesaplanabilmektedir.

Önerilen yöntem, evrişim işlemlerinden kullanılacak olan çekirdek matrislerinin ‘1’ ve ‘0’ değerlerinden oluşan matrislerin doğrusal kombinasyonu olacak şekilde bileşenlere ayrılmasına dayanmaktadır.

Şekil 3.9’da imge bulanıklaştırma evrişiminde kullanılan Gaussian bulanıklık süzgeci gösterilmektedir.

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Şekil 3.9. Gaussian bulanıklık süzgeci

Gaussian bulanıklık süzgeciyle yapılan imge bulanıklaştırma evrişimi doğrudan tümlev imgeyle hesaplanamamaktadır. Bu nedenle, Eşitlik (3.3)'ten yararlanarak alt matrislerine ayrıştırılmıştır. Şekil 3.10'da Gaussian bulanık süzgecinin alt matrislere ayrıştırılması gösterilmiştir.

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Şekil 3.10. Gaussian bulanıklık süzgecinin alt matrislere ayrıştırılması

Ayrıştırılan alt matrisler incelendiğinde her bir $G_{n \times n}^{l=i}$ çekirdek matrisinin yapısı tümlev imge yapısına uygun olduğu için, tümlev imgeyle kayıpsız ve hızlı bir şekilde evrişim yapılabilmektedir. Tümlev imgenin doğrudan uygulanamadığı fakat önerilen yöntemle hesaplanabilen çekirdek matrislerden birisi de kenar tespit evrişimi için kullanılan çekirdek matrisidir. Şekil 3.11'de kenar tespit süzgecinin alt matrislere ayrıştırılması gösterilmiştir.

$$-1 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Şekil 3.11. Kenar tespit süzgecinin alt matrislere ayrıştırılması

Tümlev imgenin doğrudan uygulanamadığı diğer bir çekirdek matrisi de keskinleştirme evrişiminde kullanılan keskinleştirme süzgeci. Keskinleştirme süzgeci alt matrislere ayrıldığında, alt matrislerin yapısının tümlev imgeyle hesaplanmaya uygun olduğu görülmektedir. Bu sayede keskinleştirme evrişimi tümlev imge ile hesaplanabilmektedir. Şekil 3.12’de keskinleştirme süzgecinin alt matrislere ayrıştırılması gösterilmektedir.

$$-1 \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Şekil 3.12. Keskinleştirme süzgecinin alt matrislere ayrıştırılması

Bu çalışmada, 5 farklı çözünürlüğe sahip imge için 8 farklı çekirdek bulanıklaştırma evrişimi yapılmıştır. Evrişimde kullanılan çekirdek matrislerinden 5×5 boyutuna sahip çekirdek matrisinin alt matrislere ayrıştırılması Şekil 3.13’te gösterilmiştir.

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 2 & 2 & 4 & 2 & 2 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Şekil 3.13. 5×5 bulanıklık süzgecinin alt matrislere ayrıştırılması

Ayrıştırılmış alt matrisler üzerinden tümlev imge yöntemiyle evrişim işlemine, giriş görüntüsüne uygun katman oluşturularak başlanmıştır. Oluşturulan katman, giriş görüntüsüne eklenmiştir. Daha sonra, katman eklenmiş giriş görüntüsünün tümlev imgesi oluşturulmuştur. Şekil 3.14’te giriş görüntüsü için tümlev imge gösterilmiştir.

4	4	4	3	7	8	3	2	2	2
4	4	4	3	7	8	3	2	2	2
4	4	4	3	7	8	3	2	2	2
3	3	3	6	4	5	7	8	8	8
5	5	5	6	4	7	2	5	5	5
5	5	5	7	4	2	6	8	8	8
0	0	0	3	2	4	6	8	8	8
1	1	1	4	9	0	5	7	7	7
1	1	1	4	9	0	5	7	7	7
1	1	1	4	9	0	5	7	7	7

0	0	0	0	0	0	0	0	0	0	0
0	4	8	12	15	22	30	33	35	37	39
0	8	16	24	30	44	60	66	70	74	78
0	12	24	36	45	66	90	99	105	111	117
0	15	30	45	60	85	114	130	144	158	172
0	20	40	60	81	110	146	164	183	202	221
0	25	50	75	103	136	174	198	225	252	279
0	25	50	75	106	141	183	213	248	283	318
0	26	52	78	113	157	199	234	276	318	360
0	27	54	81	120	173	215	255	304	353	402
0	28	56	84	127	189	231	276	332	388	444

(a)

(b)

Şekil 3.14. Giriş imgesinin tümlev imgesi (a) Giriş imgesi, (b) Giriş imgesinin tümlev imgesi

Şekil 3.13’te gösterilen ayrıştırılmış bulanıklık süzgecinin ilk alt çekirdeğinin tümlev imgeyle çözümlenmesinde kullanılacak piksel noktaları Şekil 3.15’te turkuaz rengiyle gösterilmiştir.

0	0	0	0	0	0	0	0	0	0	0
0	4	8	12	15	22	30	33	35	37	39
0	8	16	24	30	44	60	66	70	74	78
0	12	24	36	45	66	90	99	105	111	117
0	15	30	45	60	85	114	130	144	158	172
0	20	40	60	81	110	146	164	183	202	221
0	25	50	75	103	136	174	198	225	252	279
0	25	50	75	106	141	183	213	248	283	318
0	26	52	78	113	157	199	234	276	318	360
0	27	54	81	120	173	215	255	304	353	402
0	28	56	84	127	189	231	276	332	388	444

Şekil 3.15. Çekirdek matrisinin ilk parçasının tümlev imge ile çözümlenmesi

Kırmızı çerçevenin içinde kalan bölge Eşitlik (1.3)'ten yararlanılarak $(110+0)-(0+0)=110$ hesaplanmıştır. Şekil 3.13'te gösterilen ayrıştırılmış bulanıklık süzgecinin ikinci alt çekirdeğinin tümlev imgeyle çözümlenmesinde kullanılacak piksel noktaları Şekil 3.16'da turkuaz rengiyle gösterilmiştir.

0	0	0	0	0	0	0	0	0	0	0
0	4	8	12	15	22	30	33	35	37	39
0	8	16	24	30	44	60	66	70	74	78
0	12	24	36	45	66	90	99	105	111	117
0	15	30	45	60	85	114	130	144	158	172
0	20	40	60	81	110	146	164	183	202	221
0	25	50	75	103	136	174	198	225	252	279
0	25	50	75	106	141	183	213	248	283	318
0	26	52	78	113	157	199	234	276	318	360
0	27	54	81	120	173	215	255	304	353	402
0	28	56	84	127	189	231	276	332	388	444

Şekil 3.16. Çekirdek matrisinin ikinci parçasının tümlev imge ile çözümlenmesi

Burada dikkat edilmesi gereken durum ikinci alt matriste bulunan 0 değerlerinin evrişime etkisi olmadığı için boyutu 1×5 olan çekirdekle çözümlenmiştir. Eşitlik (1.3)'ten yararlanılarak $(66+0)-(44+0)=22$ hesaplanmıştır. Şekil 3.13'te gösterilen

ayrıştırılmış bulanıklık süzgecinin üçüncü alt çekirdeğinin tümlev imgeyle çözümlenmesinde kullanılacak piksel değerleri Şekil 3.17’de turkuaz rengeyle gösterilmiştir.

0	0	0	0	0	0	0	0	0	0	0
0	4	8	12	15	22	30	33	35	37	39
0	8	16	24	30	44	60	66	70	74	78
0	12	24	36	45	66	90	99	105	111	117
0	15	30	45	60	85	114	130	144	158	172
0	20	40	60	81	110	146	164	183	202	221
0	25	50	75	103	136	174	198	225	252	279
0	25	50	75	106	141	183	213	248	283	318
0	26	52	78	113	157	199	234	276	318	360
0	27	54	81	120	173	215	255	304	353	402
0	28	56	84	127	189	231	276	332	388	444

Şekil 3.17. Çekirdek matrisinin üçüncü parçasının tümlev imge ile çözümlenmesi

Burada da dikkat edilmesi gereken durum üçüncü alt matriste bulunan 0 değerlerinin evrişime etkisi olmadığı için boyutu 5×1 olan çekirdekle çözümlenmiştir. Eşitlik (1.3)’ten yararlanarak $(60+0)-(0+40)=20$ hesaplanmıştır. Şekil 3.13’te gösterilen ayrıştırılmış bulanıklık süzgecinin dördüncü ve son alt çekirdeğinin tümlev imgeyle çözümlenmesi yapılırken, çekirdek matrisinin merkez değerinin dışında kalan tüm değerler 0 olduğundan dolayı doğrudan merkez değerini temsil etmektedir. Bu yüzden giriş imgesinde bulunan değer doğrudan işleme dahil edilir. Sonuç olarak, tümlev imgeyle hesaplanan alt matrisler birleştirilerek $(110+22+20+4)=156$ sonucu elde edilmektedir. Elde edilen sonuç 36 ile bölünür ve çıkış imgesine yazılır. Giriş imgesinde bulunan her piksel için bu işlemler tekrarlanarak çıkış imgesi oluşturulmuştur. Bu sayede Şekil 3.13’te gösterilen alt matrisler üzerinden tümlev imgeyle evrişim kayıpsız bir şekilde yapılmıştır.

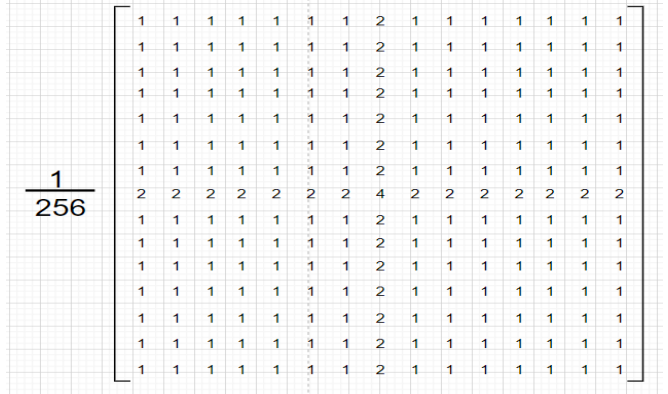
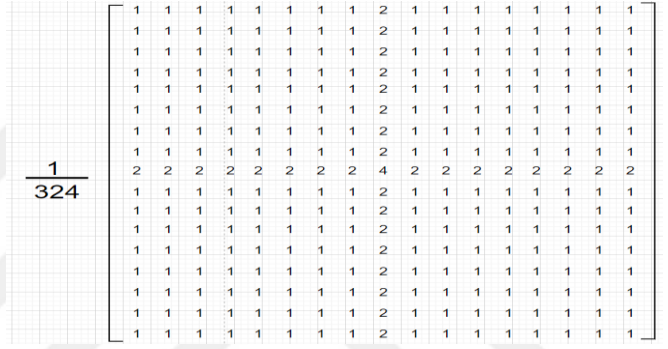
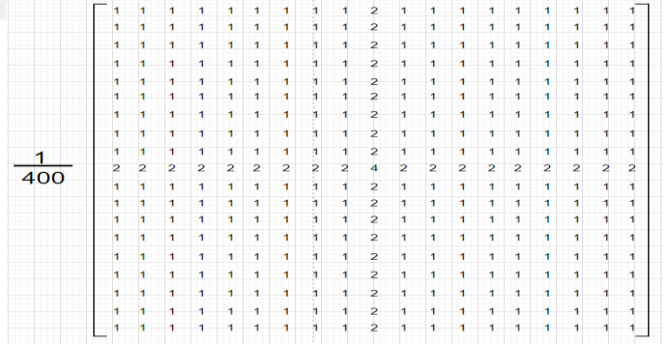
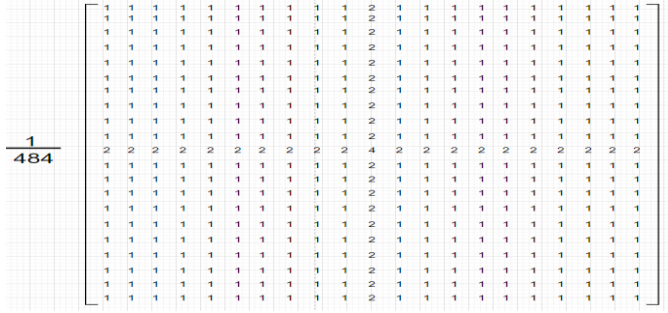
4. DENEYSEL SONUÇLAR

Bu tez kapsamında hesapsal karmaşıklığı değerlendirebilmek için çözünürlükleri 800×600 , 1280×1720 , 1440×900 , 1920×1080 ve 3840×2160 olan beş imge seçilmiştir. Bu imgeler giriş görüntüsü olarak kullanılmıştır. Farklı boyutlara sahip çekirdek matrisleriyle sırasıyla geleneksel evrişim ve tümlev imgeyle evrişim yapılmıştır. Hesapsal karmaşıklığın çözümlenmesinde ele alınan çekirdek matris boyutları, 3×3 , 5×5 , 9×9 , 15×15 , 17×17 , 19×19 , 21×21 ve 25×25 'tir. Evrişim işlemlerinde kullanılan çekirdek matrisleri Tablo 4.1'de gösterilmiştir.

Tablo 4.1. Evrişimde kullanılan çekirdek matrisleri ve boyutları

Boyut	Çekirdek Matrisi
3×3	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$
5×5	$\frac{1}{36} \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 2 & 2 & 4 & 2 & 2 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$
9×9	$\frac{1}{100} \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 4 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \end{bmatrix}$

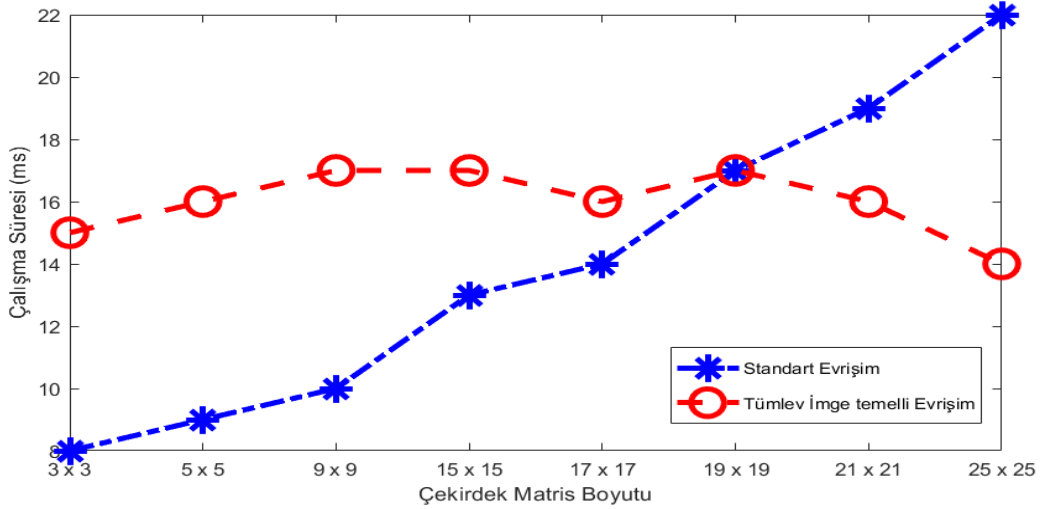
Tablo 4.1. (Devam) Evrişimde kullanılan çekirdek matrisleri ve boyutları

<p>15×15</p>	$\frac{1}{256}$ 
<p>17×17</p>	$\frac{1}{324}$ 
<p>19×19</p>	$\frac{1}{400}$ 
<p>21×21</p>	$\frac{1}{484}$ 

Tablo 4.1. (Devam) Evrişimde kullanılan çekirdek matrisleri ve boyutları

25×25	
-------	--

Şekil 4.1’de çözünürlüğü 800×600 olarak seçilen giriş imgesi için geleneksel evrişimin ve tümlev imge ile evrişimin hesaplama sürelerinin karşılaştırılması gösterilmiştir.

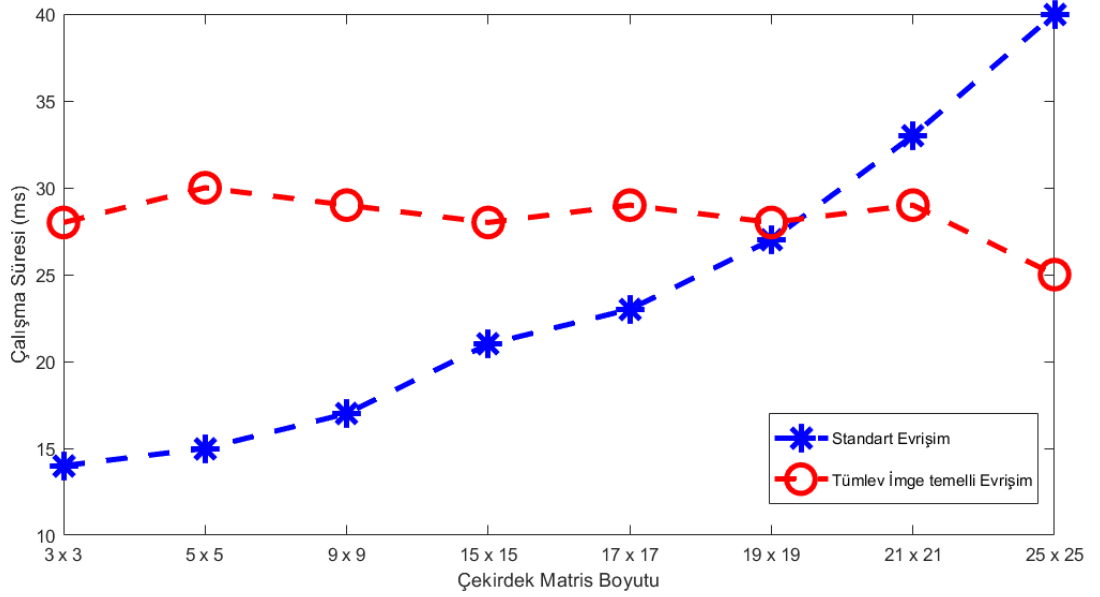


Şekil 4.1. 800×600 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması

Şekil 4.1’de görüldüğü üzere geleneksel evrişim için kullanılan çekirdek matrisinin boyutu 3×3 için 7ms’lerin sonlarında, 15×15 için 13ms’in başlarında, 21×21 için 23ms’lere yaklaştığı görülmektedir. Geleneksel evrişim için hesaplama sürelerinin tamamı sırasıyla incelendiğinde, hesaplama süresinin çekirdek matrisinin büyüklüğüne bağlı olarak arttığı gözlemlenmektedir. Tümlev imge tabanlı yapılan evrişim için hesaplama süreleri incelediğinde çekirdek büyüklüğünden bağımsız

16ms'lerde olduğu görülmektedir. 21×21 ve daha büyük boyutlara sahip çekirdek matrisleriyle gerçekleştirilen geleneksel evrişimin hesapsal karmaşıklığı, tümlev imge tabanlı evrişimin hesapsal karmaşıklığına göre daha yüksek olduğu gözlemlenmiştir.

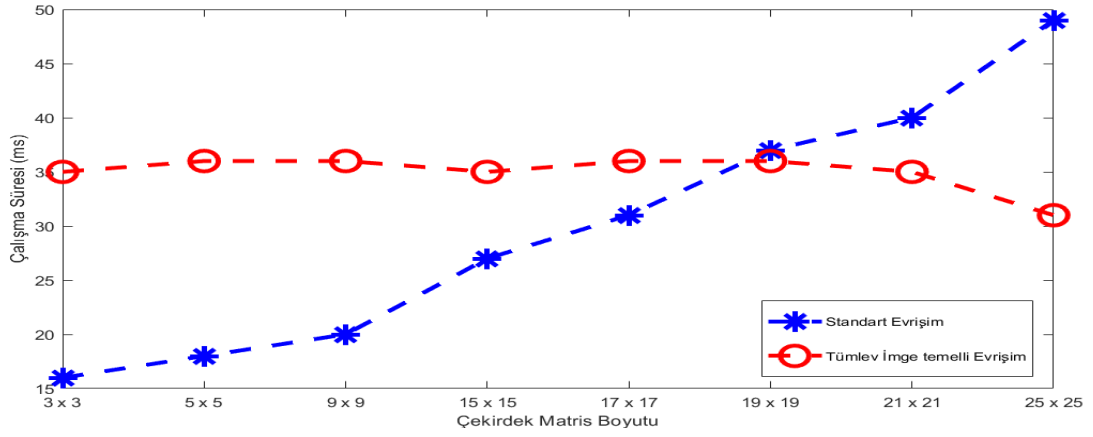
Şekil 4.2'de çözünürlüğü 1280×720 olarak seçilen giriş imgesi için geleneksel evrişimin ve tümlev imge ile evrişimin hesaplama sürelerinin karşılaştırılması gösterilmiştir.



Şekil 4.2. 1280×720 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması

Şekil 4.2'de görüldüğü üzere geleneksel evrişim için kullanılan çekirdek matrisinin boyutu 3×3 için 14ms'lerin başlarında, 15×15 için 21ms'in sonlarında, 21×21 için 33ms'lere yaklaştığı görülmektedir. Geleneksel evrişim için hesaplama sürelerinin tamamı sırasıyla incelendiğinde, hesaplama süresinin çekirdek matrisinin büyüklüğüne bağlı olarak arttığı gözlemlenmektedir. Tümlev imge tabanlı yapılan evrişim için hesaplama süreleri incelendiğinde çekirdek büyüklüğünden bağımsız 28ms'lerde olduğu görülmektedir. 21×21 ve daha büyük boyutlara sahip çekirdek matrisleriyle gerçekleştirilen geleneksel evrişimin hesapsal karmaşıklığı, tümlev imge tabanlı evrişimin hesapsal karmaşıklığına göre daha yüksek olduğu gözlemlenmiştir.

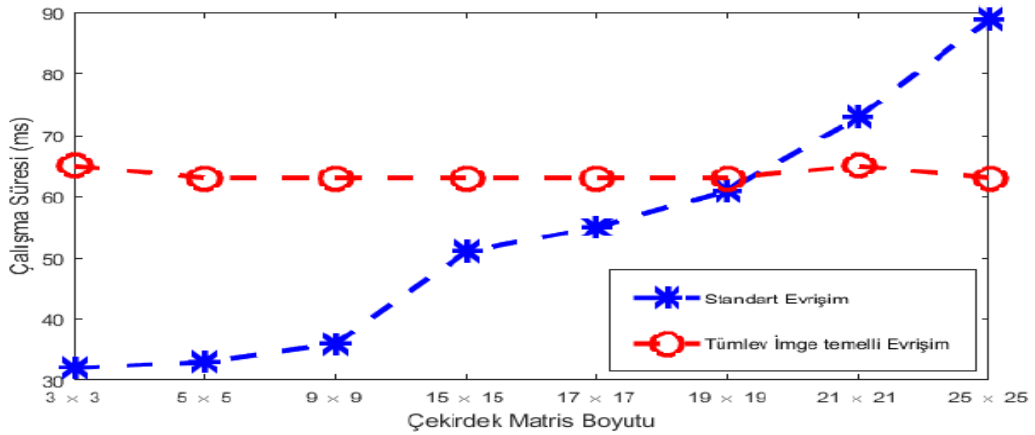
Şekil 4.3'te çözünürlüğü 1440×900 olarak seçilen giriş imgesi için geleneksel evrişimin ve tümlev imge ile evrişimin hesaplama sürelerinin karşılaştırılması gösterilmiştir.



Şekil 4.3. 1440×900 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması

Şekil 4.3'te görüldüğü üzere geleneksel evrişim için kullanılan çekirdek matrisinin boyutu 3×3 için 16ms'lerin sonlarında, 15×15 için 28ms'in başlarında, 21×21 için 41ms'lere yaklaştığı görülmektedir. Geleneksel evrişim için hesaplama sürelerinin tamamı sırasıyla incelendiğinde, hesaplama süresinin çekirdek matrisinin büyüklüğüne bağlı olarak arttığı gözlemlenmektedir. Tümlev imge tabanlı yapılan evrişim için hesaplama süreleri incelendiğinde çekirdek büyüklüğünden bağımsız 35ms'lerde olduğu görülmektedir. 19×19 ve daha büyük boyutlara sahip çekirdek matrisleriyle gerçekleştirilen geleneksel evrişimin hesapsal karmaşıklığı, tümlev imge tabanlı evrişimin hesapsal karmaşıklığına göre daha yüksek olduğu gözlemlenmiştir.

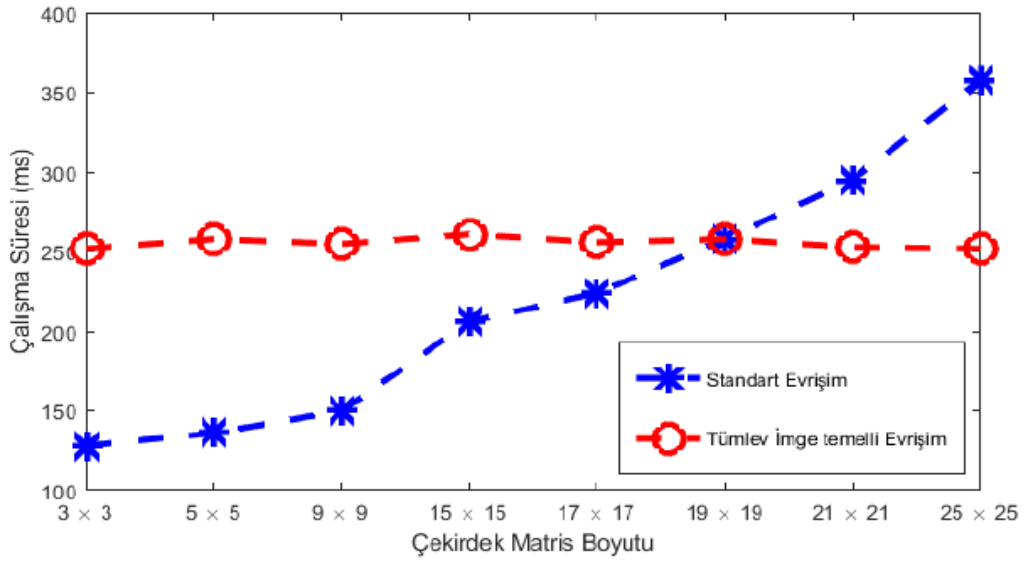
Şekil 4.4'te çözünürlüğü 2040×1080 olarak seçilen giriş imgesi için geleneksel evrişimin ve tümlev imge ile evrişimin hesaplama sürelerinin karşılaştırılması gösterilmiştir.



Şekil 4.4. 2040×1080 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması

Şekil 4.4'te görüldüğü üzere geleneksel evrişim için kullanılan çekirdek matrisinin boyutu 3×3 için 30ms'lerin başlarında, 15×15 için 50ms'in biraz üzerinde, 21×21 için 80ms'lere yaklaştığı görülmektedir. Geleneksel evrişim için hesaplama sürelerinin tamamı sırasıyla incelendiğinde, hesaplama süresinin çekirdek matrisinin büyüklüğüne bağlı olarak arttığı gözlemlenmektedir. Tümlev imge tabanlı yapılan evrişim için hesaplama süreleri incelendiğinde çekirdek büyüklüğünden bağımsız 65ms'lerde olduğu görülmektedir. 19×19 ve daha büyük boyutlara sahip çekirdek matrisleriyle gerçekleştirilen geleneksel evrişimin hesapsal karmaşıklığı, tümlev imge tabanlı evrişimin hesapsal karmaşıklığına göre daha yüksek olduğu gözlemlenmiştir.

Şekil 4.5'te çözünürlüğü 3840×2160 olarak seçilen giriş imgesi için geleneksel evrişimin ve tümlev imgeyle evrişimin hesaplama sürelerinin karşılaştırılması gösterilmiştir.



Şekil 4.5. 3840×2160 çözünürlüğe sahip giriş imgesi için geleneksel evrişim ile tümlev imge tabanlı evrişim sürelerinin karşılaştırılması

Geleneksel evrişim için kullanılan çekirdek matrisinin boyutu 3×3 için yaklaşık olarak 130ms'lerde, 15×15 için 200ms'in başlarında, 21×21 için 300ms'lerde olduğu görülmektedir. Tümlev imge tabanlı evrişim için hesaplama sürelerine bakıldığında çekirdek büyüklüğünden bağımsız 250ms civarında ufak değişimler görülmektedir. Şekil 4.1'de elde edilen sonuçlara benzer şekilde, geleneksel evrişim için hesaplama süreleri çekirdek matrisi boyutuna bağlı olarak düzenli bir şekilde arttığı, tümlev imge tabanlı evrişim işlemlerinin hesaplama süreleri çekirdek matrisinin boyutundan

bağımsız olduğu görülmektedir. Önerilen yöntem sayesinde, 19×19 ve daha büyük boyutlu çekirdek matrislerle gerçekleştirilen tümlev imge tabanlı evrişimin hesapsal karmaşıklığı, geleneksel evrişime göre daha az olduğu gözlemlenmiştir.

Farklı çekirdek matrisleri üzerinden yapılan ayırıştırma işlemleri sonucunda farklı sayıda alt çekirdek matrisi elde edilebilmektedir. Alt çekirdek matris sayısının işlem süresine etkisi, tez kapsamında ayırıştırılan çekirdek matris sayısından daha az ve daha fazla olmak üzere iki farklı tipte çekirdek matrisiyle sırasıyla geleneksel evrişim ve tümlev imge temelli evrişim yapılmış ve sonuçlar Tablo 4.2’de verilmiştir.

Tablo 4.2. Farklı sayıda alt çekirdek matrisleri için evrişim süreleri

Çekirdek Matrisi Boyutları		5×5	9×9	15×15	19×19
Fazla sayıda alt matris	Normal evrişim süresi	15,7ms	19,6ms	26,1ms	50,7ms
	Tümlev imge temelli evrişim süresi	67,7ms	67,9ms	68,7ms	68,9ms
Az sayıda alt matris	Normal evrişim süresi	16,5ms	20,1ms	27,7ms	54,1ms
	Tümlev imge temelli evrişim süresi	17,2ms	18,4ms	17,7ms	18,9ms

Tez kapsamında ayırıştırılan alt matrislerden daha fazla sayıya sahip çekirdek matrisiyle yapılan geleneksel evrişim süresinin çekirdek matrisi boyutuna bağlı olarak arttığı görülmüştür. Fakat tümlev imge temelli evrişim süresinin çekirdek süresiyle kayda değer bir ilişkisinin olmadığı görülmüştür. Tümlev imge temelli evrişim süresinin 21×21 boyutlu çekirdek matrisi dahil olmak üzere geleneksel evrişim süresinden daha fazla olduğu görülmüştür. Daha az sayıda alt matrise sahip çekirdeklerle yapılan geleneksel evrişim süresinin çekirdek matrisi boyutuna bağlı olarak arttığı,

tümlev imge temelli evriřim süresinin çekirdek matrisi boyutuna baęlı olarak ufak deęişmeler gösterdięi görölmüřtür. 5×5 ve daha büyük çekirdek matrisleriyle yapılan tümlev imge temelli evriřim süresinin geleneksel evriřim süresinden daha az olduęu görölmüřtür.

Evriřim işlemleri ve karşılařtırmalar Intel® Core i7-7700K işlemcili ve 16 GB belleęe ve Windows 10 işletim sistemine sahip bir bilgisayarda MATLAB R2018a ortamında gerçekleştirilmiřtir.



5. SONUÇLAR

Bu tez çalışmasında, beş farklı çözünürlükte giriş görüntüsü için farklı boyutlara sahip çekirdek matrisleriyle geleneksel evrişim ve tümlev imge tabanlı evrişim yapılmıştır. Evrişim işlemlerinin hesapsal karmaşıklık yükleri bulunarak karşılaştırma yapılmıştır.

Tümlev imge tabanlı evrişim işlemlerinde, birçok çekirdek matrisinin doğrudan tümlev imge gösterimiyle kullanılması mümkün değildir. Bu durumdaki çekirdek matrislerini ayrıştırarak tümlev imge tabanlı evrişim yapmak mümkündür. Şekil 3.10, Şekil 3.11 ve Şekil 3.12'de çekirdek matrislerinin ayrıştırma örnekleri gösterilmektedir. Ayrıştırılan alt matrisler üzerinden hızlı ve kayıpsız bir şekilde evrişim işlemi yapılmıştır. Evrişim işlemlerinin hesapsal yükleri karşılaştırılmıştır.

Giriş görüntüsünün çözünürlüğü artmasıyla iki evrişim yönteminin de hesaplama süreleri artmıştır. Geleneksel evrişim işlemlerinde, çekirdek matrisinin boyutu arttıkça evrişim işleminin süresinin arttığı görülmüştür. Tümlev imge tabanlı evrişim işlemlerinde, çekirdek matrislerinin boyutlarının artması, evrişim işlemlerinin süreleri üzerinde etkisinin olmadığı görülmüştür. Bu durumun seçilen beş giriş imgesi içinde geçerli olduğu gözlemlenmiştir. 800×600 ve 1280×720 çözünürlüğüne sahip imgelerde 21×21 ve daha büyük, diğer üç imgede 19×19 ve daha büyük boyutlara sahip çekirdek matrisleriyle gerçekleştirilen evrişim işlemlerinde, tümlev imge tabanlı evrişim işleminin geleneksel evrişime göre daha az hesapsal karmaşıklığa sahip olduğu gözlemlenmiştir.

KAYNAKLAR

- [1] Samtaş G., Gülesin M., Sayısal görüntü işleme ve farklı alanlardaki uygulamaları, *Ejovoc (Electronic Journal of Vocational Colleges)*, 2011, **2**(1), 85-97.
- [2] Viola P., Jones M. J., Robust Real-Time Face Detection, *International Journal of Computer Vision*, 2004, **57**(2), 137-154.
- [3] Porikli F., A Fast Way to Extract Histograms in Cartesian Spaces, *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, USA, 20-25 June 2005.
- [4] Bradley D., Roth G., Adaptive Thresholding using the Integral Image, *Journal of Graphics, GPU, and Game Tools*, 2007, **12**(2), 13-21.
- [5] Viola P., Jones M., Rapid Object Detection using a Boosted Cascade of Simple Features, *IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, USA, 8-14 December 2001.
- [6] Tuzel O., Porikli F., Meer P., Region Covariance: A Fast Descriptor for Detection and Classification, *European Conference on Computer Vision*, Berlin, Germany, May 2016.
- [7] Jiang L., Xie H. Pan B., Speeding up digital image correlation computation using the integral image technique, *Optics and Lasers in Engineering*, 2015, **65**, 117-122.
- [8] Ouyang P., Yin S., Zhang Y., Liu L., Wei S., A fast integral image computing hardware architecture with high power and area efficiency, *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2014, **62**(1), 75-79.
- [9] Demiröz B. E., Salah A. A., Bastanlar Y., Akarun L., Affordable person detection in omnidirectional cameras using radial integral channel features, *Machine Vision and Applications*, 2019, **30**(4), 645-655.
- [10] Holzer S., Rusu R. B., Dixon M., Gedikli S., Navab N., Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, October 2012.
- [11] Zhu Q., Avidan S., Yeh M. C., Cheng K. T., Fast Human Detection Using a Cascade of Histograms of Oriented Gradients, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, USA, 17-22 June 2006.

- [12] Agrawal M., Konolige K., Blas M. R., CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching, *European Conference on Computer Vision*, Munich, Germany, 8-14 October 2008.
- [13] Messom C. H., Barczak A. L., Fast and efficient rotated haar-like features using rotated integral images, *Int. J. Intelligent Systems Technologies and Applications*, 2009, **7**(1), 40-57.
- [14] Crow F. C., Summed-area tables for texture mapping, *11th annual conference on Computer graphics and interactive techniques*, New York, USA, 1984.
- [15] Bay H., Tuytelaars T., Gool L. V., SURF: Speeded Up Robust Features, *European Conference on Computer Vision*, Graz, Austria, July 2006.
- [16] Yuan F., A fast accumulative motion orientation model based on integral image for video smoke detection, *Pattern Recognition Letters*, 2008, **29**(7), 925-932.
- [17] Jung J. H., Lee H. S., Lee J. H., Park D. J., A Novel Template Matching Scheme for Fast Full-Search Boosted by an Integral Image, *IEEE Signal Processing Letters*, 2010, **17**(1), 107-110.
- [18] Zhang K., Lu J., Lafruit G., Cross-Based Local Stereo Matching Using Orthogonal Integral Images. *IEEE transactions on circuits and systems for video technology*, 2009, **19**(7), 1073-1079.
- [19] Veksler O., Fast Variable Window for Stereo Correspondence using Integral Images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, USA, 18-20 June 2003.
- [20] Facciolo G., Limare N., Meinhardt E., Integral Images for Block Matching. *Image Processing On Line*, 2014, **4**, 344-369.
- [21] Messom C., Barczak A., Stream Processing of Integral Images for Real-Time Object Detection, *2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dunedin, New Zealand, 1-4 December 2008.
- [22] Mikhisor M., Wyvill G., McCane B., Mills S., The Integral Image Method for Fisheye Images. *2013 28th International Conference on Image and Vision Computing New Zealand*, Wellington, New Zealand, 27-29 November 2013.
- [23] Wang J., Long F., Chen J., Yao J., A Novel Eye Localization Method Based on Log-Gabor Transform and Integral Image, *Applied Mathematics & Information Sciences*, 2012, **6**(2), 323-329.
- [24] Kumar J., Doermann D., Fast Rule-line Removal using Integral Images and Support Vector Machines, *2011 International Conference on Document Analysis and Recognition*, Beijing, China, 18-21 September 2011.
- [25] Huang C., Huang J., A fast HOG descriptor using lookup table and integral image, *arXiv preprint arXiv:1703.06256*, 2017.

- [26] https://en.wikipedia.org/wiki/Edge_detection, (Ziyaret tarihi: 10 Mart 2021).
- [27] Basarvarajaiah M., <https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411>, (Ziyaret tarihi: 11 Mart 2021).
- [28] Keys R. G., Cubic Convolution Interpolation for Digital Image Processing, *IEEE transactions on acoustics, speech, and signal processing*, 1981, **29**(6), 1153-1160.
- [29] Alard C., Image subtraction using a space-varying kernel, *Astronomy and Astrophysics Supplement Series*, 2000, **144**(2), 363-370.
- [30] Cabral B., Leedom L., Imaging Vector Fields Using Line Integral Convolution. *20th annual conference on Computer graphics and interactive techniques*, Anaheim, USA, 2-6 August 1993.
- [31] Meijering E. H. W., Niessen W. J., Viergever M. A., Quantitative Evaluation of Convolution-Based Methods for Medical Image Interpolation, *Medical image analysis*, 2001, **5**(2), 111-126.
- [32] Chen L. C., Papandreou G., Schroff F., Adam H., Rethinking Atrous Convolution for Semantic Image Segmentation, *arXiv preprint arXiv:1706.05587*, 2017.
- [33] Sangwine S. J., Colour image edge detector based on quaternion convolution, *Electronics Letters*, 1998, **34**(10), 969-971.
- [34] Chen L. C., Zhu Y., Papandreou G., Schroff F., Adam H., Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, *European Conference on Computer Vision*, Munich, Germany, 8-14 September 2018.
- [35] Sangwine S. J., Ell T. A., Colour image filters based on hypercomplex convolution. *Vision, Image and Signal Processing*, 2000, **147**(2), 89-93.
- [36] Chen L. C., Papandreou G., Kokkinos I., Murphy K., Yuille A. L., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **40**(4), 834-848.
- [37] Li B., Acton S. T., Active Contour External Force Using Vector Field Convolution for Image Segmentation. *IEEE transactions on image processing*, 2007, **16**(8), 2096-2106.
- [38] Zhou D., Shen X., Dong W., Image zooming using directional cubic convolution interpolation, *IET Image Processing*, 2012, **6**(6), 627-634.
- [39] Fan Y., Yu J., Liu D., Huang T. S., Scale-Wise Convolution for Image Restoration, *AAAI Conference on Artificial Intelligence*, New York, USA, 7-12 February 2020.

KİŞİSEL YAYINLAR VE ESERLER

Tunçel T., Akbulut O., Ayrılabilir Çekirdekler ile Tümlev İmge temelli Hızlı Evrişim İşlemleri, *29. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Türkiye, 9-11 Haziran 2021.



ÖZGEÇMİŞ

Lise öğrenimini Hacı Fatma Erdemir Anadolu Lisesi'nde tamamladı. 2012 yılında girdiği Karabük Üniversitesi Bilgisayar Mühendisliğini 2018 yılında tamamladı. 2018 yılında Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. 2021, Haziran ayında mezun olma durumundadır.

