

Görüntü Filtre Çekirdek Matrisinin Genetik Algoritmalar ile Eğitiminin Bir Analizi

An Analysis of Genetic Algorithm with Training of Image Filter Kernel Matrix

Süleyman UZUN¹, Devrim AKGÜN²

¹Bilgi İşlem Daire Başkanlığı, Bilecik Şeyh Edebali Üniversitesi

suleyman.uzun@bilecik.edu.tr

²Bilgisayar Mühendisliği, Sakarya Üniversitesi

dakgun@sakarya.edu.tr

Özet

Sunulan çalışmada, konvolüsyon veya korelasyon ile gerçekleştirilen doğrusal görüntü filtreleme işlemlerinde kullanılan çekirdek matrisinin genetik algoritmalar ile eğitimi analiz edilmiştir. Bu amaçla, filtreleme başarımının filtre ağırlıklarını içeren çekirdek matrisinin boyutu ve görüntü boyutuna bağlı olarak değişimi incelenmiştir. Deneysel çalışmalarda, farklı boyutlarda görüntü ve her boyut için farklı örnek görüntüler kullanılmıştır. Gürültülü görüntüler Gaussian gürültüsü eklenerek oluşturulmuştur. Değiştirilen parametrelerin etkisini görmek için, hesaplamalar, genetik algoritmaların popülasyon adedi ve iterasyon sayısı sabit olacak şekilde elde edilmiştir. Elde edilen sonuçlara göre, aynı filtre çekirdeği kullanıldığında görüntü boyutlarının değişiminin hesaplama kalitesine etkisi düşük seviyelerde ortaya çıkmıştır. Görüntü filtre çekirdeğinin boyutu artırıldıkça, filtreleme kalitesi düştüğü gözlemlenmiştir. Genetik algoritmanın değişken adedi artmasıyla oluşan bu durum, iterasyon adedinin artırılmasıyla iyileştirilmektedir.

Abstract

In the presented study, training of kernel matrix which is used for convolution or correlation image filtering operations using genetic algorithms was analyzed. For this purpose, filtering performance was analyzed according to the size of kernel matrix and the size of the image. In the experimental studies, various sizes of images and various example images for each size were used. Noisy images were formed by adding Gaussian noise. Computations were obtained for fixed number of population and iterations to see the effect of parameters. According to obtained results, the effect of image size variation is at small levels when the same filtering kernel is used. Performance degradation was observed as the size of the kernel matrix is increased. This degradation which is due to the increase in the number of variables of genetic algorithm is improved by increasing the number of iteration.

1. Giriş

Görüntü işleme teknikleriyle gürültü yok etme, belirli ayrıntıları ortaya çıkartma, görüntü yumuşatma, kenar bulma, vb. birçok işlem gerçekleştirilebilir [1-6]. Görüntü filtreleme tekniklerinden doğrusal konvolüsyon veya benzer bir işlem

olan korelasyonlar yapısındaki filtreler, bilimsel ve endüstriyel uygulamalarda yaygın olarak kullanılmaktadır. Filtrelemede kullanılan çekirdekler pratikte 3×3 boyutundan başlayıp 21×21 gibi yüksek boyutlara ulaşabilen ağırlıklar matrisidir. Çekirdek matrislerinin ağırlık değerleri kullanılan filtreleme amacına göre değişmektedir [1], [7]. Filtre çekirdek matrisi ağırlıkları çeşitli analitik yöntemlerle hesaplamalar sonucu hazır olarak belirlenebilir [8]–[10]. Diğer bir yaklaşımda filtreler orijinal ve gürültü örnekler kullanılarak çekirdek ağırlıkları eğitilebilir [11-14]. Bunun için orijinal ve gürültülü örnekler arasındaki farka bakılarak elde edilen hata sinyali kullanılır. Çeşitli optimizasyon yöntemleriyle hatayı sıfıra yaklaştıran en uygun çekirdek matrisi belirlenebilir. Bu amaçla genetik algoritmalar gibi sezgisel yöntemlere günümüzde yaygın olarak başvurulmaktadır [15-17]. Genetik algoritmalarda rastgele oluşturulan ilk popülasyondaki her bir birey kullanılarak uygunluk değerleri hesaplanır. Bu uygunluk değerlerine göre yeniden yeni popülasyonlar oluşturulur. Bu işlemler durdurma şartına; yani sınır olarak belirtilen uygunluk değerine kadar devam eder. İşlemlerin en sonunda gürültülü görüntüyü en iyi şekilde gürültüden ayıran ağırlıklar belirlenir.

Sunulan çalışmada çekirdek matrisi ağırlıkları orijinal ve gürültülü görüntüler kullanılarak genetik algoritmalar ile hesaplanmıştır. Bu amaçla orijinal test görüntülerine Gaussian gürültüsü eklenerek gürültülü görüntüler elde edilmiş ve genetik algoritmalar ile eğitimde kullanılmıştır. Deneysel sonuçlar, farklı boyutlardaki görüntüler için, popülasyon adedi ve iterasyon sayısı sabit olacak şekilde karşılaştırmalı olarak elde edilmiştir.

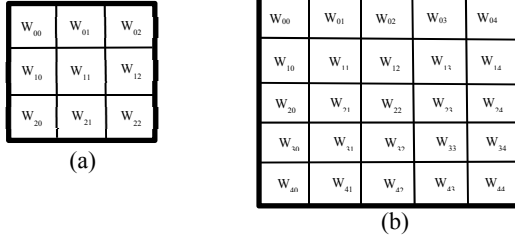
2. Doğrusal Görüntü Filtreleme

Doğrusal görüntü filtreleme teknikleri olan konvolüsyon veya korelasyon işlemlerinde, filtre çekirdek matrisinin giriş görüntüsünün tüm pikselleri üzerinde gezdirilmesiyle filtreleme işlemi gerçekleştirilir. Bu işleme ait sık kullanılan matematiksel ifade eşitlik 1’de görülmektedir [1].

$$y(i, j) = \sum_{m=-N}^N \sum_{n=-N}^N w_{m,n} x(i-m, j-n) \quad (1)$$

Denklemden “ $y(i, j)$ ” çıkış görüntüsünü, “ $x(i-m, j-n)$ ” giriş görüntüsünü ve “ $w_{m,n}$ ” filtre ağırlıklarını göstermektedir.

Görüntünün i ve j pikselleri için tanımlanmış bu işlem tüm pikselleri için gerçekleştirilir. Giriş görüntüsü piksel değerleri ile filtre çekirdeğinin ağırlıkları bir birleriyle çarpılıp sonunda bütün çarpımlar toplanarak yeni piksel değeri hesaplanır. Örneğin; Şekil 1'de 5×5 ve 3×3 boyutlarında iki farklı filtre çekirdeği görülmektedir. Şekil 1.a' da 3×3 boyutlarına sahip filtre çekirdeği 9 ağırlığa sahip iken, Şekil 1.b' de 5×5 boyutlarına sahip filtre çekirdeği ise 25 ağırlığa sahiptir.



Şekil 1 (a) 3×3 Boyutlarına sahip filtre çekirdeği (b) 5×5 boyutlarına sahip filtre çekirdeği genel görünümü.

Şekil 2'de Laplasyan filtresi ile filtreleme işlemine bir örnek gösterilmektedir. Filtre çekirdeğinin merkezi, giriş görüntüsü üzerinde hesaplanması gereken pikselin üzerine gelecek şekilde yerleştirilir ve eşitlik 2'de gösterildiği gibi filtre çekirdeğinin her bir pikselinin değeri ile giriş görüntüsüne denk gelen pikseller bir birleriyle çarpılarak toplanır.

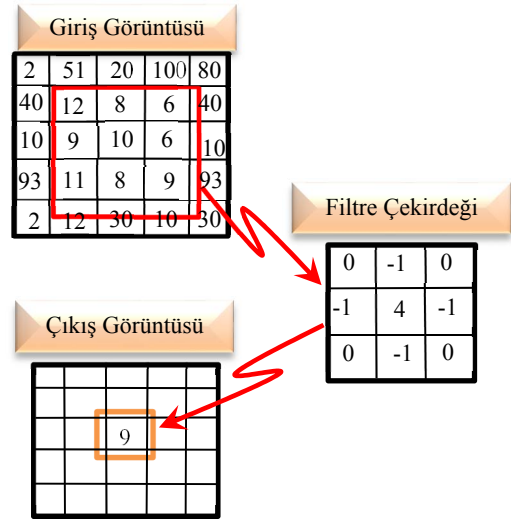
$$y = 12*0 + 8*(-1) + 6*0 + 9*(-1) + 10*4 + 6*(-1) + \dots \quad (2)$$

$$\dots 11*0 + 8*(-1) + 9*0 = 9$$

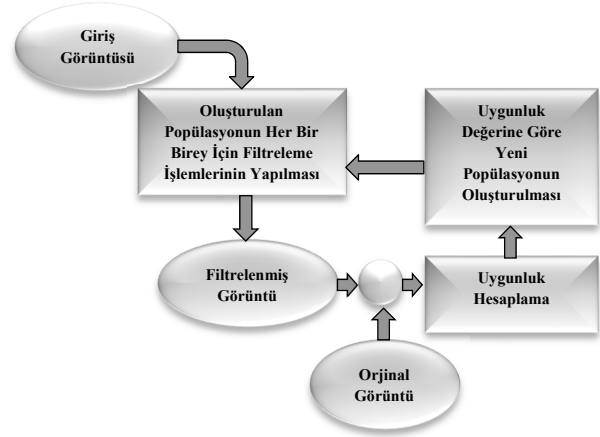
Elde edilen sonuç filtre çekirdeğinin merkezinin üzerinde bulunduğu pikselin yeni değeri olmaktadır. Bu şekilde filtre çekirdeği, giriş görüntüsü üzerinde dolaştırılarak çıkış görüntüsünün bütün pikselleri yeniden hesaplanmaktadır. Yapılan bu çalışmada renkli görüntüler kullanılmıştır. Renkli görüntüler Gaussian gürültüsü eklenerek filtreleme işlemleri yapılmaktadır.

3. Genetik Algoritmalar

Genetik algoritmalar doğal seçim ve doğal genetik yapıya sahip arama ve optimizasyon yöntemidir [17], [18]. Buna benzer algoritmalar çözümü elde etmek için çaprazlama ve mutasyon operatörlerini kullanırlar. Genetik algoritmalarda elde edilen her bir çözüme kromozom ya da birey adı verilmektedir. Elde edilen çözümlerden en iyilerinin belirlenerek bir sonraki nesle aktarılmasını işlemine seçme işlemi denilmektedir. Genetik algoritmalar sonucunda rastgele birden çok çözüm elde edebilirler. Elde edilen bu çözümler arasından en iyileri seçilerek bir sonraki nesle aktarılırlar ve kötü sonuca sahip olanlar ise elenmektedirler. Böylece aktarılan en iyi çözümler zamanla yeni nesillerinde ortaya çıkmasını sağlamaktadırlar. Genetik algoritmalarla filtre çekirdeği ağırlıkları hesaplanırken bütün örnek görüntüler için iterasyon sayısı 200, popülasyon sayısı 100, mutasyon oranı 0.05 ve çaprazlama oranı 0.3 olarak belirlenmiştir. Şekil 3'de genetik algoritmanın çalışma şeması görülmektedir. Sistem girişinde renkli görüntü üzerine Gaussian gürültüsü eklenerek genetik algoritmalarla gönderilmektedir. Genetik algoritmalar rastgele ilk popülasyon oluşturularak bu gürültülü görüntüyü filtreler ve orijinal görüntüden çıkartarak uygunluk (fitness) değeri hesaplar.



Şekil 2 Konvolüsyon ile filtreleme örneği.



Şekil 3 Genetik Algoritmaların Çalışma Diyagramı.

Hesaplanan bu uygunluk değerlerine göre yeni popülasyonlar üretilir ve işlemler tekrarlanır. Bu tekrarlamalar yapılan bu çalışma için genetik algoritmaları sonlandırma kriteri olarak iterasyon sayısı kullanıldığından her bir görüntü için 200 defa tekrarlanmaktadır. Sonuçta elde edilen en iyi uygunluk değerine göre filtre çekirdeği ağırlıkları bulunmaktadır. Bu ağırlıklar kullanılarak gürültülü görüntü son olarak filtrelenip ekranda gösterilmektedir.

4. Deneysel Sonuçlar ve Tartışmalar

Yapılan çalışmalar Windows 7 Home Premium™ 64-bit işletim sistemi, Intel™ Core 2 Duo 2.13GHz işlemci, 4GB Ram belleğe sahip bilgisayar üzerinde yapılmıştır. Geliştirilen algoritma VC++ programlama dilinde yazılmıştır. Deneysel 256×256, 512×384, 512×512 ve 1024×1024 piksel boyutlarına sahip her birinden beş farklı görüntü kullanılmıştır [19]. 3×3 , 5×5 ve 7×7 matris boyutlarına sahip filtre çekirdeği kullanılmıştır. Görüntülere Gaussian gürültüsü eklendiğinde gürültülü görüntünün MSE (Mean Squared Error) değeri 26,37'dir. Bütün görüntülerde popülasyon adeti 100 ve maksimum iterasyon sayısı 200 olarak seçilmiştir.

Çizelge 1: 3×3 Filtre çekirdeği için MSE değerleri.

3×3 Filtre Çekirdeği İçin Ölçülen MSE Değerleri						
Görüntü Boyutları	Görüntü-1	Görüntü-2	Görüntü-3	Görüntü-4	Görüntü-5	Ortalama
256×256	9,29	9,47	9,78	11,34	8,84	9,75
512×384	10,27	8,91	14,44	10,38	9,41	10,68
512×512	8,88	14,28	9,85	10,80	9,40	10,64
1024×1024	10,57	12,64	12,85	10,25	9,96	11,25

Çizelge 2: 5×5 Filtre çekirdeği için MSE değerleri.

5×5 Filtre Çekirdeği İçin Ölçülen MSE Değerleri						
Görüntü Boyutları	Görüntü-1	Görüntü-2	Görüntü-3	Görüntü-4	Görüntü-5	Ortalama
256×256	12,40	14,67	13,05	14,43	13,41	13,59
512×384	12,91	8,75	19,44	14,88	9,96	13,19
512×512	9,00	18,32	9,89	13,54	12,26	12,60
1024×1024	12,47	15,38	14,13	13,18	13,73	13,78

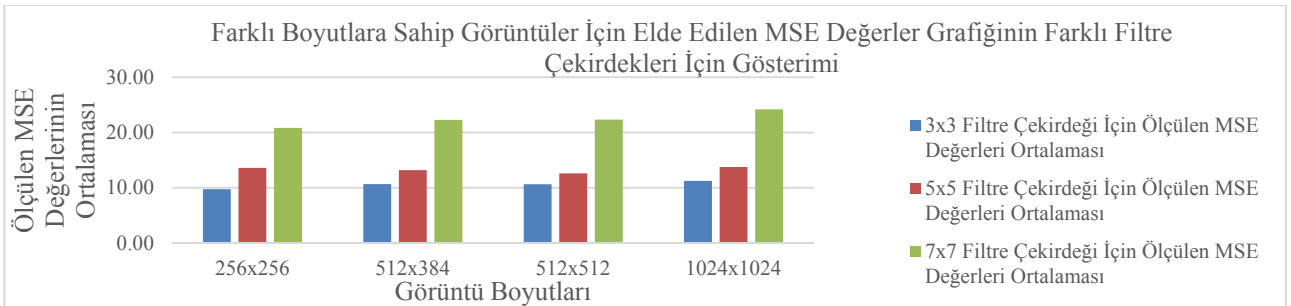
Çizelge 3: 7×7 Filtre çekirdeği için MSE değerleri.

7×7 Filtre Çekirdeği İçin Ölçülen MSE Değerleri						
Görüntü Boyutları	Görüntü-1	Görüntü-2	Görüntü-3	Görüntü-4	Görüntü-5	Ortalama
256×256	22,18	21,24	22,06	20,97	17,79	20,85
512×384	23,88	18,93	28,00	15,14	25,51	22,29
512×512	19,13	27,93	18,01	24,91	21,76	22,35
1024×1024	26,85	22,65	25,99	22,07	23,39	24,19

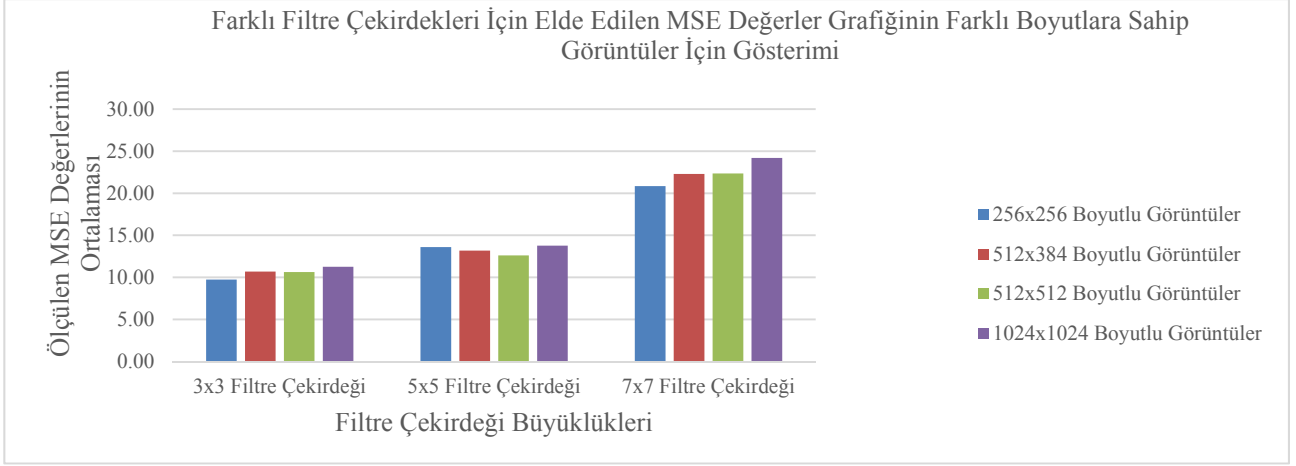
Çizelge 1, Çizelge 2 ve Çizelge 3'te çeşitli boyutlardaki beş farklı görüntü için sırasıyla 3×3, 5×5 ve 7×7 boyutlarında filtre çekirdekleri kullanılarak hesaplanan MSE değerleri görülmektedir. Matris boyutları arttıkça filtreleme sonrasında elde edilen MSE değerleri artmaktadır. Filtre çekirdeği matrisinin boyutlarının artması, iterasyon ve popülasyon sayılarının sabit tutulması durumunda elde edilen filtreleme kalitesini düşürmektedir. Bunun sebebi ise genetik algoritmaların eğitmesi gereken değişken adedinin artmasıdır. Bundan dolayı pratikte iterasyon sayısının yeterince artırılması gerekir.

Ortalama MSE değerlerinin görüntü ve çekirdek boyutuna bağlı değişimleri değişimi sırasıyla Şekil 4 ve Şekil 5'de daha net bir şekilde görülmektedir. Şekil 4'de aynı görüntü boyutuna sahip görüntülerin farklı filtre çekirdekleriyle filtrelandıklarında ortalama MSE değerlerinin birbirinden farklı oldukları daha iyi görülmektedir. Bu farklılığın sebebi sabit iterasyon ve popülasyon adedi kullanıldığında bunun yanı sıra filtre çekirdeğinin boyutlarının artması sonucudur. Daha iyi sonuçların elde edilebilmesi için popülasyon sayısı ile iterasyon sayılarının artırılması gerektiği görülmektedir.

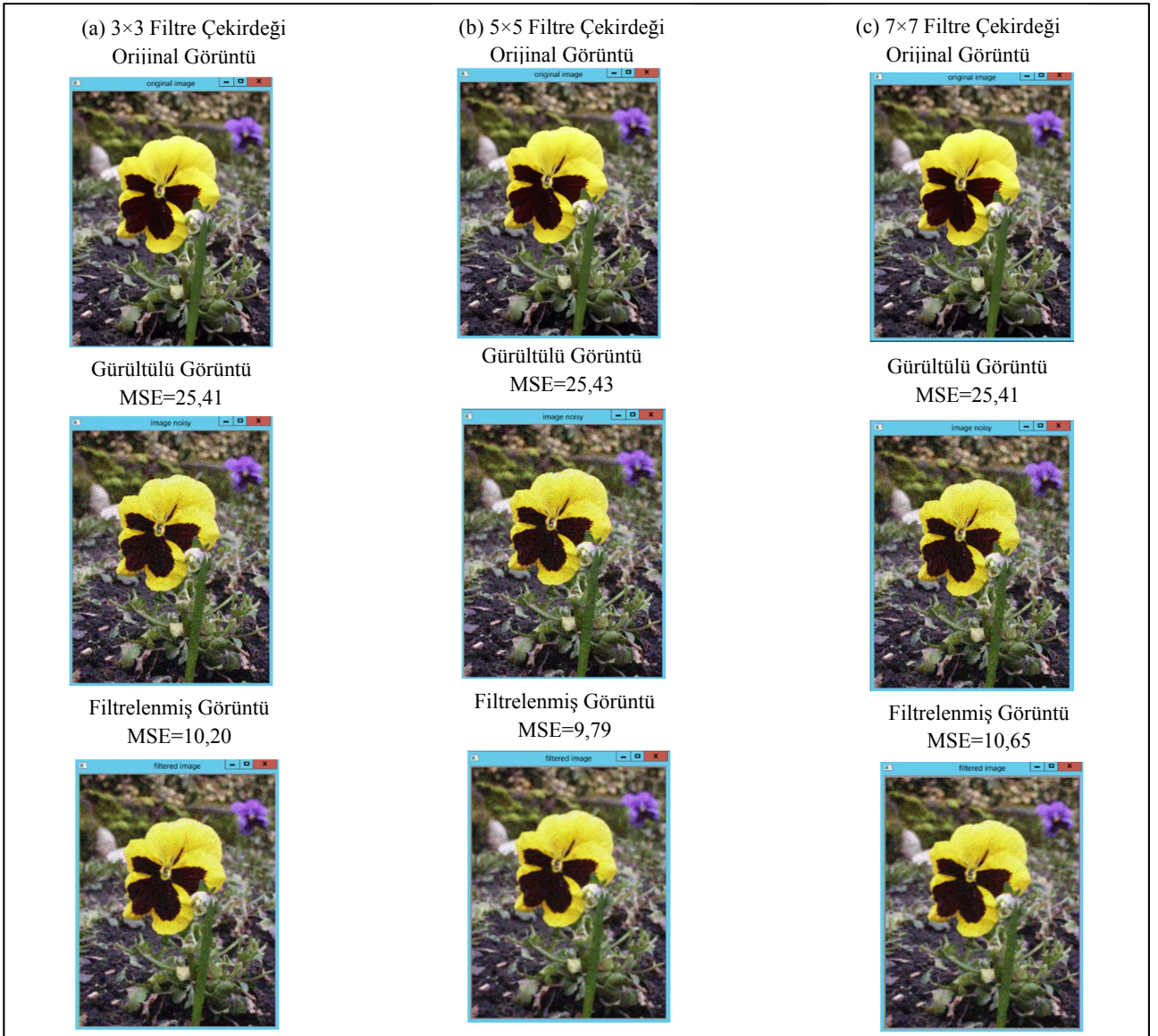
Şekil 5'de ise aynı filtre çekirdeklerinde farklı görüntü boyutlarına sahip görüntülerin filtreleme sonrasında elde edilen MSE değerlerinin birbirine çok yakın oldukları daha iyi görülmektedir. Görüntü boyutları farklı olsa da aynı filtre çekirdeği matris boyutları için değişken adedi aynı olduğundan sonuçlar benzer elde edilmektedir. Şekil 6'da UCID veri tabanlarından alınmış bir görüntü farklı filtre çekirdekleri kullanılarak filtrelenmiş örnekleri gösterilmektedir. Şekil 6.a'da 3×3 filtre çekirdeği kullanılarak filtrelenmiş görüntünün orijinal hali, gürültü eklenmiş hali ve filtrelenmiş hali görülmektedir. Gürültülü görüntü MSE değeri orijinal görüntü ile gürültülü görüntünün farkı alınarak elde edilen değerdir. Filtrelenmiş görüntü ise genetik algoritmalar tarafından elde edilen en iyi ağırlıkları kullanılarak gürültülü görüntünün filtrelenmiş halidir. Şekil 6b'de ise 5×5 çekirdek matrisi kullanılarak filtrelenmiş görüntü, görüntünün orijinal hali ve gürültülü görüntüleri görülmektedir. Şekil 6c'de ise 7×7 filtre çekirdeği kullanılarak filtrelenmiş görüntü, görüntünün orijinal hali ve gürültülü görüntüleri görülmektedir. Örnek görüntülerin her biri için genetik algoritmalarda iterasyon sayısı 400 ve popülasyon adedi 200 olarak belirlenmiştir.



Şekil 4: Farklı boyutlara sahip görüntüler için elde edilen MSE değerler grafiğinin farklı filtre çekirdekleri için gösterimi.



Şekil 5: Farklı filtre çekirdekleri için elde edilen MSE değerler grafiğinin farklı boyutlara sahip görüntüler için gösterimi.



Şekil 6: UCID veri tabanından alınmış örnek görüntüler [20].

5. Sonuçlar

Sunulan çalışmada görüntü filtre çekirdeği matrisi ağırlıklarının genetik algoritmalar kullanılarak eğitimi analiz edilmiştir. Deneysel sonuçlarda göre, popülasyon adedi 100 ve iterasyon sayısı 200 ile sabit tutulduğu zaman görüntü boyutu arttıkça aynı boyutlu filtre çekirdekleri için filtreleme kalitesi görüntü boyutundan bağımsız değişmektedir. Bunun sebebi, genetik algoritmanın başarımının kullanılan değişken adedine bağlı olmasıdır. Bunun yanında, aynı boyuttaki ve aynı filtre çekirdeği için görüntülerin farklı başarımlar sergilemesi görüntü içeriği ile ilişkisinden kaynaklanmaktadır. Deneysel çalışmalar iterasyon sayısı sabit tutulup görüntü filtre çekirdeğinin büyüklüğü açısından da ele alındığında, çekirdek boyutunun artması filtre başarımını düşürmüştür. Bunun sebebi, örneğin; 3×3 filtre çekirdeği için genetik algoritmanın 9 değişkeni eğitmesi gerekirken, 5×5 için değişken adedinin 25'e çıkması olarak gösterilebilir. Değişken adedinin artması ile düşen performansı dengelemek için iterasyon sayısı veya popülasyon adedi artırılarak kalite iyileştirilebilir.

6. Kaynaklar

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. 2007.
- [2] D. Ziou and S. Tabbone, "Edge detection techniques - An overview," *Int. J. Pattern Recognit. Image Anal.*, vol. 8, no. 4, pp. 1–41, 1998.
- [3] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [4] F. Kiraç and L. Akarun, "Parçacık süzgeci kullanarak görüntü dizisinde insan eli izleme," in *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference, SIU 2005*, 2005, vol. 2005, pp. 561–564.
- [5] D. Akgün, "A practical parallel implementation for TDLMS image filter on multi-core processor," *J. Real-Time Image Process.*, Jan. 2014.
- [6] D. Akgün, "Paralel Görüntü Filtreleme İçin Çok Çekirdekli Bilgisayar Üzerinde Başarım Analizi," *İleri Teknol. Bilim. Derg.*, 2013.
- [7] V. Hlavac, "Fundamentals of Image Processing," in *Optical and Digital Image Processing: Fundamentals and Applications*, 2011, pp. 71–96.
- [8] A. Karasaridis and E. Simoncelli, "A filter design technique for steerable pyramid image transforms," *1996 IEEE Int. Conf. Acoust. Speech, Signal Process. Conf. Proc.*, vol. 4, pp. 2387–2390, 1996.
- [9] J. Yang, L. Liu, T. Jiang, and Y. Fan, "A modified Gabor filter design method for fingerprint image enhancement," *Pattern Recognit. Lett.*, vol. 24, no. 12, pp. 1805–1817, 2003.
- [10] P. Zahradnik, B. Šimák, and M. Vlček, "Filter design for image preprocessing in image communication," in *Proceedings of the 8th International Conference on Networks, ICN 2009*, 2009, pp. 40–45.
- [11] R. Poli, "Genetic Programming for Image Analysis," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, 1996, pp. 363–368.
- [12] D. Akgün and P. Erdoğan, "GPU accelerated training of image convolution filter weights using genetic algorithms," *Appl. Soft Comput.*, vol. 30, pp. 585–594, 2015.
- [13] D. J. Krusienski and W. K. Jenkins, "Particle swarm optimization for adaptive IIR filter structures," *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 1, pp. 965–970 Vol.1, 2004.
- [14] S. K. Saha, S. Mukherjee, D. Mandal, R. Kar, and S. P. Ghoshal, "Gravitational search algorithm in digital FIR low pass filter design," *2012 Third Int. Conf. Emerg. Appl. Inf. Technol.*, no. i, pp. 52–55, 2012.
- [15] M. Haseyama and D. Matsuura, "A filter coefficient quantization method with genetic algorithm, including simulated annealing," *Signal Process. Lett. IEEE*, 2006.
- [16] W. Chang, "Coefficient estimation of IIR filter by a multiple crossover genetic algorithm," *Comput. Math. with Appl.*, 2006.
- [17] G. J. E. Rawlins, "Foundations of Genetic Algorithms," in *Foundations of Genetic Algorithms*, 1991, vol. 21, p. 341.
- [18] K. de Jong, "Learning with Genetic Algorithms: An Overview," *Mach. Learn.*, vol. 3, no. 2, pp. 121–138, 1988.
- [19] G. Weber, "USC-SIPI image database: Version 4," 1993.
- [20] G. Schaefer and M. Stich, "UCID - An Uncompressed Colour Image Database," *SPIE, Storage Retr. Methods Appl. Multimed.*, vol. 5307, pp. 472–480, 2003.