Görüntü İşleme Dersi
Proje Ara Rapor Formu

“Pixera”

Doğukan Mehmet ALADAĞLI - 223301001

Başak GÖRGÜLÜ - 223301051

Mislina KEKLİK - 223301048

Safa KELKİT- 223301024

1. Giriş

Hazırlamış olduğumuz ve ismini Pixel ile Era kelimelerinin birleşiminden alan 'Pixera' adlı projemiz kapsamında, Python programlama dili kullanarak gri dönüşüm, binary dönüşüm, görüntü döndürme, görüntü kırpma, görüntü yakınlaştırma ve uzaklaştırma, renk uzayı dönüşümleri, giriş görüntüsüne ait histogram ve orjinal görüntü histogramını germe/genişletme, iki resim arasında ekleme ve bölme gibi aritmetik işlemler, kontrast artırma, konvolüsyon işlemi (mean), tek eşikleme, kenar bulma algoritmalarının kullanımı (prewitt), görüntüye gürültü ekleme (Salt& Pepper) ve filtrelerin kullanımı (mean, median) ile gürültü temizleme, görüntüye filtre uygulanması (Unsharp) ve genişleme, aşınma, açma, kapama gibi işlemleri görüntü üzerinde uygulayabilmek amacıyla yazmış olduğumuz fonksiyonları C# Windows Forms Application ile tasarlamış olduğumuz görsel arayüze entegre ederek elimizdeki görüntü üzerinde yukarıda bahsedilen görüntü işleme işlemlerini yapma olanağı sağlayan bir masaüstü uygulaması geliştirmek amaçlanmıştır.

2. Literatür Taraması

Görüntü işleme, bilgisayarların görsel verileri analiz edip anlamlandırmasını sağlayan geniş kapsamlı bir bilim dalıdır. Literatürde bu alanda gerçekleştirilen çalışmalar; dijital görüntülerin iyileştirilmesi, filtrelenmesi, sınıflandırılması, nesne tespiti gibi birçok alt disiplini kapsamaktadır. Bu projede ele alınan temel görüntü işleme teknikleri, bu alanın başlangıç düzeyinde en sık kullanılan ve algoritmik olarak en iyi anlaşılması gereken yöntemlerini içermektedir.

Gri renk dönüşümüyle ve renk uzayı dönüşümü ile ilgili olarak özellikle medikal alanlarda güvenliğe yönelik yapılan belli başlı çalışmalarda araştırmacılar renkli bir görseli görüntü işleme tekniklerini kullanarak gri tonlamaya dönüştürürken, orjinal renk bilgisini geri kazanılabilir şekilde gizlemeyi ve görüntünün zarar görme durumlarının tespitini sağlamayı hedeflemişlerdir. Gri dönüşüm sürecinde orjinal yani renkli görüntü CIELAB renk uzayına dönüştürülmüştür. Renk bilgisi yani a ve b kanalları, oluşturulan gri görüntünün LSB (least significant bit)'inde saklanmış ve sıkıştırma işlemleri gerçekleştirilmiştir. Bu görüntüye filigran eklenmiştir. Bu sıkıştırma işlemleri sonucunda görüntünün bozulup bozulmadığının kontrolü için gizlenen bilgiler geri getirilmiş ve kontrol işlemleri gerçekleştirilmiştir. Bu çalışma özellikle medikal alanı gibi belli başlı uygulamalarda kullanılabilir duruma getirilmiştir. (8)

Başka bir çalışmada ise projeye uygun olarak tek eşikleme, binary dönüşüm, morfolojik işlemler gibi konular ele alınmıştır. Bu çalışmada araştırmacılar gri bir görüntüyü belirlenen bir eşik değeri kullanarak binary görüntüye, yani siyah-beyaz formata dönüştürmeyi amaçlamışlardır. Eşik seçilirken histogram analizi dikkate alınmıştır. Nesne ve arka plan görüntülerinin net ayrılabildiği bimodal histogramlar kullanılmıştır. Tek eşikleme işlemi, karmaşık dokulara sahip görüntülerde çoklu eşikleme yöntemine göre başarısız olsa da proje kapsamında bu yöntem ele alınmaktadır. Genişleme, aşınma, açma ve kapama gibi morfolojik işlemler ele alınmıştır. Bu işlemlerle beraber nesneler büyütülerek küçük delikler kapatılmış, sonrasında nesneler küçültülerek ince çıkıntılar temizlenmiş, açma işlemi ile küçük nesneler ve gürültüler ortadan kaldırılmış ve kapama işlemi ile de nesne içerisindeki delikler doldurulmuştur. (9)

Histogram germe/genişletme işlemlerinin anlaşılması konusunda incelenen başka bir çalışmada ise düşük kontrastlı görüntülerin netleştirilmesi amacıyla histogram germe/genişletme işlemi uygulanmıştır. Bu çalışmada histogram germe düşük kontrastlı görüntülerin parlaklık değerlerini genişleterek görüntüyü iyileştirmek amacıyla kullanılmıştır. Çalışmada kullanılan ve aşağıda yer alan parçalı fonksiyonla beraber belirli değer aralığında bulunan bölgelerde iyileştirme yapılmıştır.

$$s = \begin{cases} l * r & 0 \le r < a \\ m * (r - a) + v & a \le r < b \\ n * (r - b) + w & b \le r < L - 1 \end{cases}$$

Çalışmada l,n = 0.5 ve m = 2.0 olarak alınmıştır. Alt eşik (a) ve üst eşik (b) değerleri ise a = 50 ve b = 200 olarak belirlenmiştir. Yukarıdaki parçalı fonksiyona göre 0 <= r < 50 değer aralığındaki pikseller 0.5 çarpanı ile beraber daha koyu hale, 200 <= r <= 255 değer aralığındaki pikseller 0.5 çarpanı ve w = 225 değeri ile beraber daha parlak hale getirilmiştir. 50 <= r < 200 değer aralığındaki piksellerin ise kontrastı artırılarak görüntü iyileştirilmeye çalışılmıştır. (10)

Kenar bulma algoritmalarının kıyaslanmasına yönelik olan çalışmada Sobel ve Prewitt algoritmalarının farklarına değinilmiştir. Prewitt ve Sobel operatörleri 3x3 konvolüsyon maskeleri (bir görüntü üzerinde matematiksel işlemler yapmak için kullanılan matrisler) kullanarak görüntülerin X ve Y gradyanlarını (Gx ve Gy) belirlemiştir. Aşağıda yer alan tabloda Prewitt operatörü içerisinde kullanılan 3x3 konvolüsyon maskeleri yer almaktadır.

-1	0	+1
-1	0	+1
-1	0	+1

Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

Konvolüsyon maskeleri kullanılarak tespit edilen gradyanlar görüntünün yoğunluğundaki değişimin büyüklüğünü ve yönünü ifade etmek için kullanılmıştır. Gradyan büyüklüğü |G| = |Gx| + |Gy| formülüyle yaklaşık bir değer olarak elde edilmiştir. Bu bulunan gradyan büyüklüğü, kenarın ne kadar keskin olduğunu gösterir. Prewitt operatörün Sobel operatöründen farkı, spektral tepkisinin daha basit düzeyde olmasıdır. Spektral tepki, kullanılan maskelerin veya filtrelerin farklı frekans bileşenlerine nasıl tepki verdiğini göstermektedir. Daha basit düzeyde bir spektral tepkiye sahip olması sebebiyle Prewitt operatörü, yüksek gürültülü görüntülerde Sobel operatörüne göre daha başarısız olmuştur. Bu proje kapsamında kullanılması gereken Prewitt operatörü ise Sobel operatörüne göre bu bağlamda daha hızlı hesaplamalar yapabilmiştir. (5-11)

Salt & Pepper gürültüsü görüntü işleme yöntemlerinin kullanılması sırasında görüntü verilerine eklenen bir gürültüdür. Test ve değerlendirme amacıyla kullanılmaktadır. Görüntüye siyah (0) ve beyaz (255) pikseller ekleyerek görüntüye gürültü eklenmesini amaçlar. Bu proje kapsamında yer alan mean veya median görüntü temizleme yöntemlerinin test edilmesi için kullanılabilecek yöntemlerden biridir (14-16). İncelenen çalışmada RGB görüntüleri ele alınmıştır. “Salt and Pepper” gürültüsünün etkileri incelenerek median filtresi kullanımının ne derecede başarılı olabileceği hesaplanmıştır. Eklenen gürültü oranı arttırıldıkça orjinal görüntüde bulan renk kombinasyonlarının tekrar sayıları önemli ölçüde değişmiştir. Bu gürültünün temizlenmesi için ele alınan çalışmada median filtresi kullanılmıştır. Orjinal RGB görüntü R (kırmızı), G (yeşil) ve B (blue) kanallarına ayrılmış ve her kanal bapımsız olarak

ele alınmıştır. Median filtresini uygularken 3x3 boyutunda bir maske kullanılmıştır. Bu maske görüntü üzerinde piksel piksel kaydırılarak her adımda pencerede bulunan 9 piksel değeri sıralanmış ve bunların medyanı alınmıştır. Bu hesaplama sonucunda "Salt and Pepper" gürültüsü uygulanan pikseller bulunmuş ve bu pikseller içerisinde bulunduğu pencerenin medyan değeriyle değiştirilmiştir. Son olarak işlemin başında bağımsız olarak ayrılan R (kırmızı), G (green) ve B (mavi) renk kanalları birleştirilerek gürültüsüz yeni görüntü elde edilmiştir. Filtreleme işlemlerinin değerlendirilmesi için kullanılan PSNR ve RMSE metriklerinin sonuçları aşağıda listelenmiştir.

Noise ratio	PSNR	RMSE
0.001	140.2946	0.0033
0.005	130.5774	0.008
0.01	125.3474	0.0146
0.015	120.8481	0.021
0.02	118.1216	0.027
0.025	115.4155	0.0345
0.03	113.897	0.0408
0.035	111.7628	0.0478
0.04	110.0766	0.0534
0.045	109.36	0.0592
0.05	108.2835	0.0678
0.055	107.0779	0.0749
0.06	104.6116	0.0848
0.065	103.3155	0.0929
0.07	103.2653	0.0988
0.075	101.1348	0.106
0.08	99.4528	0.115
0.085	99.1635	0.117
0.09	97.8145	0.1335

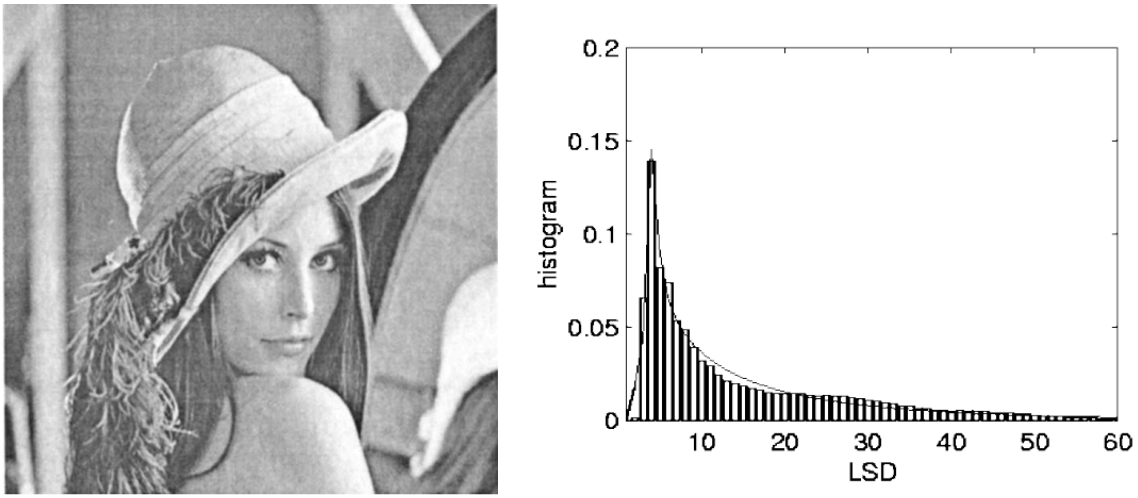
(13)

- PSNR (Peak Signal-to-Noise Ratio): Orjinal görüntü ve filtrelenmiş görüntü arasındaki benzerlik derecesini ölçer. Bu değer ne kadar yüksek olursa filtrelenmiş görüntü orjinal görüntüye o kadar yakındır. (13)

- RMSE (Root Mean Square Error): Orjinal görüntü ve filtrelenmiş görüntü arasındaki farkı ölçer. Bu değer ne kadar düşük olursa filtrelenmiş görüntü orjinal görüntüye o kadar yakındır. (13)

Bu tabloya göre elde edilen çıkarımlar, median filtresinin “Salt and Pepper” gürültüsünü temizlemek için başarılı bir yöntem olduğunu ortaya çıkarmıştır. (13)

Özellikle tıbbi görüntülerde (mesela MR) kontrast artırma işlemleri önemli bir rol oynamaktadır. Bu bağlamda ele alınan çalışma ACE (Automatic Color Enhancement) algoritmasına alternatif bir algoritma geliştirerek ACE algoritmasının yol açtığı aşırı gürültü bozulması ve kenar bozulmaları gibi sorunlara çözüm bulmayı hedeflemiştir. Bunun için yerel standart sapma (LSD - Local Standart Deviation) kullanılmıştır. Görüntülerin yerel standart sapmaları matematiksel olarak modellenmiştir. Aşağıda bu çalışmada (17) yer alan örnek bir görüntü ve bunun histogramı verilmiştir.



Oluşturulan bu LSD histogramı en küçük kareler yöntemiyle optimize edilmiştir. Kontrast kazancı fonksiyonu tanımlanmış ve bu fonksiyon düşük LSD’li bölgelerde gürültüyü, yüksek LSD’li bölgelerde ise aşırı vurgulamayı azaltmıştır. Bu çalışma tıbbi görüntü verilerinde kontrast artırma için başarılı sonuçlar ortaya koymuştur. (17)

Görüntü kalitesinin iyileştirilmesi sürecinde görüntü netliğini artırmak için kullanılan yöntemlerden biri de Unsharp Masking algoritmasıdır. Unsharp Mask algoritması, görüntünün kenarlarını vurgulayarak keskinliği artırır ve görüntüye netlik kazandırır. Ele alınan çalışmada Unsharp Mask tekniği kullanılmış ve zayıf kaldığı noktaların giderilmesi amacıyla adaptif bir Unsharp Mask Filtreleme yöntemi önerilmiştir. Kontrastı düşük görüntülerde renkleri netleştirebilmek için ilk aşamada her bir renk kanalı (R, G, B) “min” ve “max” değerlerine göre daha geniş bir aralığa ayrılmıştır. Böylece parlak ve karanlık bölgeler daha net e anlaşılabilir hale gelmiştir. Sonrasında görüntünün içerisinde yer alan gölge ve ışık durumlarının farkını ortaya çıkaran 3x3 bir kenar tespit filtresi kullanılarak görüntü içerisinde yer alan ani parlaklık değişimleri ve kenarlar tespit edilmiştir. Bu noktada araştırmacılar klasik Unsharp Mask yönteminin eksikliklerini gidermek amacıyla keskinleştirme yönteminde bir değişikliğe gitmiştir. Klasik Unsharp Mask yönteminde görüntünün keskinleştirilmesi aşamasında hep aynı şiddette bir vurgulama (gain) uygulanırken, araştırmacılar bu önerdikleri yeni Unsharp Mask yönteminde farklı bir yol izlemişlerdir. Her pikselin parlaklık ve kenar

şiddetine göre o piksele özel bir vurgulama (gain) belirlenmiştir. Bu sayede over-range (piksel değerlerinin izin verilen aralığın dışına çıkması) ihtimali önemli ölçüde azaltılmaya çalışılmış ve oluşabilecek artefaktlar önlenmiştir. 200 adet 400x300 görüntüden oluşan bir veri setiyle hazırlanan bu çalışmada hedefi doğrultusunda başarılı olmuştur. Klasik Unsharp Mask yöntemi sonucunda over-range oranı yaklaşık 0.9 iken önerilen bu yeni Adaptif Unsharp Mask yöntemi ile beraber bu oran yaklaşık olarak 0.2 değerine düşürülmüş ve bu sayede görüntüde oluşabilecek artefaktlar önemli ölçüde azaltılmıştır. Aynı şekilde entropi değerleri de görüntünün normal halinde ortalama 7.34, klasik Unsharp Mask filtresi uygulamış halinde ortalama 7.43 iken bu yeni yöntemle beraber entropi değeri ortalama 7.56 seviyesine ulaşmıştır. (18)

3. Kullanılan/Kullanılacak Teknolojiler

Bu projede görüntü işleme süreçlerini etkin bir şekilde gerçekleştirebilmek için çeşitli yazılım ve donanım teknolojileri kullanılmaktadır. Kullanılan teknolojiler, kullanıcı deneyimi ve uygulama performansı göz önünde bulundurularak seçilmiştir.

3.1 C# Windows Forms Application

Kullanıcı dostu, kullanımı kolay ve modern bir arayüz tasarlayabilmek amacıyla bu proje kapsamında arayüz tasarımı için windows forms application tercih edilmiştir. Windows forms application' a ait sürükle-bırak (drag & drop) desteği sayesinde düğme, resim kutusu, menü gibi arayüz bileşenlerini hızlıca tasarlayabilmek ve görsel öğeler üzerinde kod yazmadan işlem yapabilmek çok daha kolay hale gelmiştir. Ayrıca arayüz tasarımı için ek olarak HTML, CSS veya Javascript bilgisi gerektirmez. Python kodlarını entegre etmek kolaydır. Kullanıcıdan parametre almak (örneğin: eşik değeri) gibi işleri kutucuklarla rahatça yapabilme; PictureBox, TextBox, ComboBox gibi kontrollerle kullanıcıya görsel sonuçları kolayca sunabilme gibi imkanlar tanır. Sağladığı kolaylıklara ek olarak görüntü üzerinde yapılacak işlemler sonrasında görüntüyü hemen "PictureBox.Image" ile göstermek yeterlidir.(12)

3.2 Python Programlama Dili

Python programlama dili diğer programlama dillerine nazaran daha az satır kodla daha çok şey yapabilmemize olanak tanımıştır. Geliştirme sürecinde kolaylıklar sağlar. Yazılan her fonksiyonu hemen test edebilme imkanı tanır. Bu sayede kod, görüntü, hata ve düzeltme döngüsü çok hızlı şekilde ilerlemiş olur. Ayrıca Python komut satırından çağrıldığında gayet iyi çalışan bir programlama dilidir. Dışarıdan parametre alabilir ve C#'tan çok kolay yönetilebilir. Bu sayede python programlama dilinde yazılan görüntü işleme fonksiyonlarının arayüz tasarımını gerçekleştireceğimiz windows forms application ortamına entegre edilmesi kolay olacaktır.(12)

4. Uygulama Çalışma Prensibi

Bu proje kapsamında geliştirilen masaüstü uygulama, temel görüntü işleme algoritmalarının Python diliyle sıfırdan kodlanması ve bu algoritmaların C# Windows Forms Application ortamına entegre edilmesi prensibine dayanmaktadır. Uygulama, kullanıcıya görsel bir arayüz üzerinden farklı görüntü işleme işlemleri gerçekleştirme olanağı sunmaktadır. Her bir işlem, arka planda ilgili Python modülünün çağrılmasıyla gerçekleştirilir. Uygulamanın genel çalışma prensibi aşağıdaki adımlarla açıklanabilir.(6)

4.1. Görüntü Yükleme ve Giriş Formatı

Kullanıcı, arayüz üzerinden bilgisayarında bulunan bir görüntü dosyasını (genellikle JPEG veya PNG formatında) uygulamaya yükler. Yüklenen görüntü, arayüzde ön izlenebilir hâle getirilir ve geçici bir klasöre kaydedilir. Bu görüntü, işlem yapılmak üzere Python modülüne gönderilecektir. Bu sırada sistem(7):

· Görüntünün formatını kontrol eder (JPEG, PNG, BMP)

· Geçici çalışma dizinine kopyalar

· Arayüzde bir önizleme sağlar

4.2 İşlem Seçimi ve Parametre Girişi

Kullanıcı, arayüzdeki işlem listesinden gerçekleştirmek istediği görüntü işleme fonksiyonunu seçer. Uygulama, seçilen işleme göre kullanıcıdan ek parametreler (örneğin eşik değeri, filtre boyutu, kernel matrisi gibi) isteyebilir. Örneğin gri dönüşüm işlemi parametresiz olarak gerçekleştirilebilir ancak binary eşikleme yapacağımız zaman bir eşik değeri girmemiz gerekir veya Salt&Pepper gürültü ekleyeceğimiz zaman ise yoğunluk yüzdesi girilmesi gerekir. Bu veriler geçici olarak saklanır ve Python modülüne aktarılmak üzere hazırlanır.(1-5-3)

4.3. Python Modüllerinin Yapısı

Python tarafında her bir görüntü işleme işlemi için ayrı fonksiyonlar yazılmıştır. Bu fonksiyonlar hazır kütüphane kullanılmadan, temel matris işlemleri ve piksel manipülasyonları ile gerçekleştirilmiştir.(1-2-3-5)

· **Gri Dönüşüm:** Görüntüdeki her pikselin RGB bileşenleri belirli bir katsayıyla ağırlıklandırılarak tek kanal haline getirilmesidir.

· **Histogram Germe:** Görüntü histogramı normalize edilerek daha geniş bir aralığa yayılır, kontrast artırılmasıdır.

· **Kenar Bulma (Prewitt):** Görüntünün yatay ve dikey türevleri ayrı ayrı alınarak kenar bilgileri çıkarılmasıdır.

· **Salt&Pepper Gürültü Ekleme:** Rastgele seçilen piksellere maksimum (beyaz) ve minimum (siyah) değer atanarak görüntüye bozulma uygulanmasıdır.

4.4 İşlem Sonucunun Alınması

Python işlemi tamamlandıktan sonra, C# tarafı belirtilen dizindeki sonuç görüntüsünü alır ve arayüzde kullanıcıya gösterir. Sistem aynı zamanda: İşlem süresini ölçer, sonuç ile orijinal görüntüyü karşılaştırmalı gösterme opsiyonu sunar, "Yeniden işlem yap" veya "Yeni görüntü yükle" seçeneklerini aktif hale getirir.(1)

4.5. C# ile Python Arasında İletişim

C# arayüzü üzerinden kullanıcı bir işlem seçtiğinde, uygulama şu işlemleri yapar: ilk olarak kullanıcının seçtiği işlem türü ve yüklediği görüntünün yolu, komut satırı argümanlarıyla bir Python modülüne iletilir. Python modülü, ilgili fonksiyonu çalıştırarak işlem sonucunu yeni bir görüntü olarak kaydeder. C# tarafı bu çıktıyı okuyarak kullanıcıya sonuç görüntüsünü arayüzde gösterir. Yeni bir görüntü işleme işlemi ekleneceği zaman, yalnızca ilgili Python fonksiyonu ve C# tarafında bu fonksiyonu tetikleyen buton güncellenmektedir.(2)

4.6. Arayüz Kullanılabilirliği ve Kullanıcı Etkileşimi

Arayüzde her işlem için ayrı butonlar veya menüler tasarlanmıştır. Kullanıcı hangi işlemi gerçekleştirmek isterse, ilgili butona tıklayarak işlemi başlatabilmelidir. Görüntü üzerinde yapılan işlemler sırasında kullanıcıdan ek parametre (örneğin eşik değeri, filtre boyutu gibi) alınması gerekiyorsa, bu bilgiler windows forms arayüzünde açılan giriş kutuları aracılığıyla alınır ve Python modülüne parametre olarak iletilir.(2)

4.7. Hata Yönetimi ve Koşullar

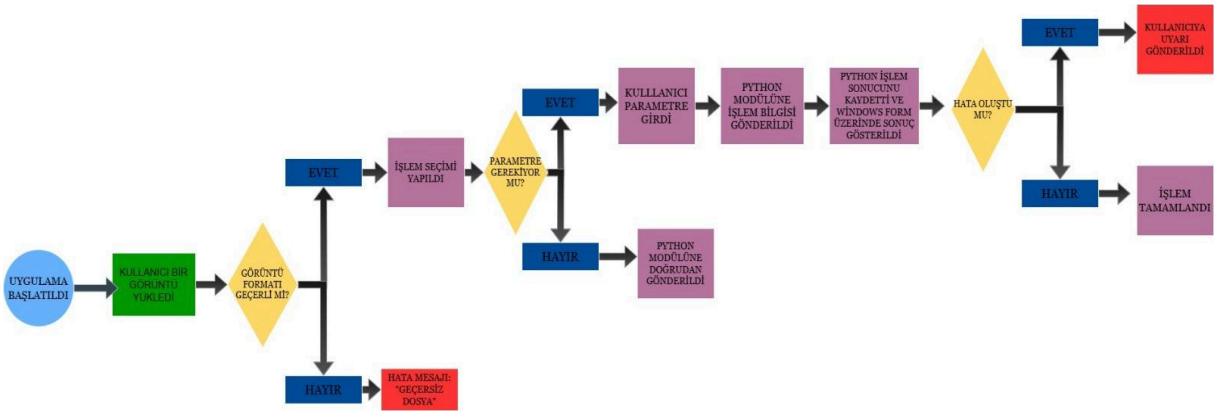
Bu sistem; her aşamada hatayı erken tespit etmeyi ve kullanıcıyı yönlendirmeyi amaçlar. Uygulama içerisinde farklı kontrol noktaları ve bazı hata mesajları yer alır (7):

· Geçersiz dosya türü yüklendiğinde uyarı verir.

· Python modülü çalışmazsa veya sonuç üretilemezse hata mesajı gösterir.

· Kullanıcının girdiği parametreler geçersizse işlem durdurulur.

Bu sistemin uygulama prensibine ait akış şeması görseli aşağıda verilmiştir:



Akış Şeması

Kaynaklar

1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Pearson Education.

2. Jain, A. K. (1989). Fundamentals of Digital Image Processing. Prentice-Hall.

3. Soille, P. (2003). Morphological Image Analysis: Principles and Applications (2nd ed.). Springer.

4. Burger, W., & Burge, M. J. (2016). Digital Image Processing: An Algorithmic Introduction Using Java (2nd ed.). Springer.

5. Prewitt, J. M. S. (1970). Object Enhancement and Extraction. In B. S. Lipkin & A. Rosenfeld (Eds.), Picture Processing and Psychopictorics (pp. 75-149). Academic Press.

6. Python Software Foundation. (2024). The Python Standard Library - sys module. https://docs.python.org/3/library/sys.html

7. PIL (Python Imaging Library) Unofficial Handbook. (2023). Image processing with PIL. https://pillow.readthedocs.io

8. Yu, F. S. K., Chan, Y., Lam, K. K. M., Lun, D. P. K. (2023). “Self-embedding reversible color-to-grayscale conversion with watermarking feature”. Signal Processing: Image Communication, 119, 117061.

9. Bovik, A. C. (2009). “Chapter 4: Basic Binary Image Processing”. The Essential Guide to Image Processing, 69-96.

10. Negi, S. S., Bhandari, Y. S., (2014). “A Hybrid Approach to Image Enhancement using Contrast Stretching on Image Sharpening and the analysis of various cases arising using Histogram”. IEEE International Conference on Recent advances and Innovations in Engineering, 1-6.

11. Amer, G. M. H., Abushaala, A. M., (2015). “Edge detection methods”. 2015 2nd World Symposium on Web Applications and Networking, 1-7.

12. C# kütüphanesinin Python’a entegre edilmesi. https://www.codeporting.com/tr/blog/integrating_a_c_sharp_library_into_python_wrapping_vs_code_conversion

13. Al-Azzeh, J., Zahran, B., Alqadi, Z., (2018). “Salt and Pepper Noise: Effects and Removal”. International Journal on Informatics Visualization, 2(4), 252-256.

14. Farzana, F. M. A., Arshadh, H., Safreen, S., Muthukumaran, N., (2018). “Design and Analysis for Removing Salt and Pepper Noise in Image Processing”. Quarterly International Journal, 3, 42-47.

15. Erkan, U., Gokrem, L., Enginoglu, S. (2018), “Different applied median filter in salt and pepper noise”. Computers and Electrical Engineering, 70, 789-798.

16. Tuz ve Biber Gürültüsü (Salt and Pepper Noise). 2008

https://bilgisayarkavramlari.com/2008/12/20/tuz-ve-biber-gurultusu-salt-and-pepper-noise/

17. Chang, D., Wu, W. (1998), “Image Contrast Enhancement Based on a Histogram Transformation of Local Standard Deviation”. IEEE Transactions on Medical Imaging, 17, 518-532.

18. Lin, S. C. F., Wong, C. Y., Jiang, G., Rahman, M. A., Ren, T. R., Kwok, N., Shi, H., Yu, Y., Wu, T. (2016). “Intensity and edge based adaptive unsharp masking filter for color image enhancement”. Optik, 127, 407-414