

Dijital Görüntü İşleme Teknik Raporu

1. Projeye Genel Bakış

Bu proje, Python tabanlı kapsamlı bir dijital görüntü işleme uygulamasıdır. **Tkinter** kullanarak grafiksel kullanıcı arayüzü (GUI) oluşturulmuş ve çeşitli görüntü işleme algoritmaları entegre edilmiştir. Uygulama, kullanıcıların farklı görüntü manipülasyonlarını kolayca gerçekleştirmesini sağlayan modüller ve kullanıcı dostu bir yapı sunar.

2. Kütüphaneler

2.1 Kullanılan Ana Kütüphaneler

- Tkinter**: Grafiksel kullanıcı arayüzü (GUI) oluşturma ve yönetimi için Python'un standart kütüphanesi.
- NumPy**: Yüksek performanslı sayısal işlemler ve özellikle matris operasyonları için temel bir kütüphane. Görüntü verilerinin verimli bir şekilde işlenmesinde kritik rol oynar.
- OpenCV (cv2)**: Bilgisayarlı görü ve görüntü işleme algoritmaları için endüstri standardı bir kütüphane. Karmaşık görüntü analizleri ve dönüşümleri için geniş fonksiyon desteği sunar.
- PIL (Python Imaging Library - Pillow)**: Görüntü manipülasyonu, format dönüşümleri ve Tkinter ile görüntülerin entegrasyonu için kullanılır.
- Pandas**: Veri analizi ve özellikle nesne tespiti gibi işlemlerden elde edilen sonuçların Excel çıktısı olarak dışa aktarılması için kullanılır.

2.2 Kütüphane Seçim Gerekçeleri

- Tkinter**: Python'ın yerleşik GUI kütüphanesi olması, ek kurulum gerektirmemesi ve hızlı prototipleme imkanı sunması nedeniyle tercih edilmiştir.
- OpenCV**: Güçlü görüntü işleme algoritmaları (örn. Hough Dönüşümü, nesne tespiti) ve geniş fonksiyon desteği sayesinde karmaşık işlemleri kolayca uygulamaya olanak tanır.
- PIL**: Çeşitli görüntü formatları arasında dönüşüm yapabilme yeteneği ve Tkinter'ın **PhotoImage** sınıfı ile uyumluluğu nedeniyle seçilmiştir.
- NumPy**: Görüntülerin matrisler olarak temsil edildiği dijital görüntü işlemede yüksek performanslı hesaplamalar için vazgeçilmezdir.

3. Uygulama Mimarisi

3.1 Ana Sınıf Yapısı

Uygulamanın tüm işlevselliği, **ImageProcessingApp** adlı ana bir sınıf içinde kapsüllenmiştir. Bu sınıf, GUI bileşenlerini, veri yapılarını, görüntü işleme algoritmalarını ve yardımcı fonksiyonları düzenli bir şekilde bir araya getirir.

3.2 Veri Yapıları

Uygulama, farklı görüntü formatlarını desteklemek ve algoritmaların gereksinimlerine göre esneklik sağlamak amacıyla çoklu veri yapılarını kullanır:

- **original_image_data**: Ham görüntü verisini özel bir **TXT formatında** saklar. Bu format, piksel verilerine doğrudan ve kolay erişim sağlar, özellikle geometrik dönüşümler için idealdir.
- **original_image_pil**: Orijinal görüntüyü **PIL (Pillow)** formatında tutar. Bu, farklı görüntü formatlarından yükleme ve görüntülerin GUI'de gösterimi için kullanılır.
- **processed_image_data**: İşlenmiş görüntü verisini yine özel **TXT formatında** depolar. Bu, işlenmiş görüntünün daha sonra kaydedilmesi veya diğer geometrik işlemlere tabi tutulması için kullanılır.
- **processed_image_pil**: İşlenmiş görüntüyü **PIL formatında** tutar. İşlenmiş görüntünün GUI'de gösterimi için gereklidir.
- **current_image_cv**: Mevcut görüntüyü **OpenCV formatında (RGB, height × width × 3 boyutlarında bir NumPy dizisi)** saklar. OpenCV tabanlı işlemler (örn. S-Curve, Hough, Deblurring) bu format üzerinde çalışır.
- **object_features_data**: Nesne tespiti işlemlerinden elde edilen özellikleri (örn. konum, boyut) depolamak için bir liste yapısıdır. Bu veriler daha sonra Excel çıktısı için kullanılır.

4. Kullanıcı Arayüzü Tasarımı

4.1 Ana Düzen

Uygulama, fonksiyonelliği ve kullanıcı deneyimini optimize etmek için **ikili panel yaklaşımı** kullanır:

- **Sol Panel (400px genişlik)**: Kontrol panelleri, parametre giriş alanları ve işlem seçeneklerini içerir. Bu panel, kullanıcı etkileşimlerinin merkezidir.
- **Sağ Panel**: Orijinal ve işlenmiş görüntülerin yan yana gösterildiği ana görüntüleme alanıdır. Bu düzen, kullanıcıların uygulanan işlemlerin sonuçlarını anında karşılaştırmasına olanak tanır.

4.2 Görüntü Gösterim Alanları

- **Orijinal Görüntü Çerçevesi**: Sol tarafta konumlandırılmıştır ve yüklenen görüntüyü gösterir.
- **İşlenmiş Görüntü Çerçevesi**: Sağ tarafta konumlandırılmıştır ve uygulanan işlemlerin sonucunu gösterir. Her iki alan da **dinamik boyutlandırma** özelliğine sahiptir, yani büyük görüntüler ekran alanına sığacak şekilde otomatik olarak küçültülür ve oransal koruma sağlanır.

4.3 Kontrol Paneli Organizasyonu

Modüler tasarım prensibi ile kontrol paneli beş ana kategoriye ayrılmıştır, bu da kullanıcıların işlemleri kolayca bulmasına ve seçmesine yardımcı olur:

- **Dosya İşlemleri**:
 - TXT formatında görüntü yükleme
 - Normal görüntü formatlarını (JPG, PNG vb.) yükleme
 - İşlenmiş görüntüyü TXT veya diğer standart formatlarda kaydetme
- **Ana İşlem Kategorileri**:
 - **Geometrik Dönüşümler**: Boyutlandırma, zoom, döndürme gibi temel geometrik manipülasyonlar.

- **S-Curve Kontrast Güçlendirme:** Görüntülerin kontrastını iyileştirmek için sigmoidal fonksiyonlar kullanır.
- **Hough Transform:** Çizgi ve diğer basit geometrik şekillerin tespiti.
- **Deblurring:** Görüntüdeki bulanıklığı giderme.
- **Nesne Sayma ve Özellik Çıkarma:** Belirli nesneleri tespit etme ve bunlar hakkında istatistiksel bilgi çıkarma.

5. İşlem Kategorileri Detaylı Analizi

5.1 Geometrik Dönüşümler

Bu bölüm, görüntülerin boyutunu, konumunu veya oryantasyonunu değiştiren işlemleri kapsar. Uygulama, hem özel TXT formatı üzerinde doğrudan çalışarak piksel hassasiyetini korur hem de farklı interpolasyon yöntemleri sunarak kalite kontrolü sağlar.

Desteklenen İşlemler:

- **Büyütme (`resize_larger`):** Görüntüyü belirtilen ölçek faktörü kadar büyütür.
- **Küçültme (`resize_smaller`):** Görüntüyü belirtilen ölçek faktörü kadar küçültür.
- **Zoom In/Out işlemleri:** Görüntünün belirli bir merkez etrafında yakınlaştırılması veya uzaklaştırılması.
- **Döndürme (`rotate`):** Görüntüyü belirtilen açı kadar döndürür.

İnterpolasyon Yöntemleri:

Görüntü yeniden boyutlandırma veya döndürme gibi işlemlerde yeni piksel konumlarına düşen değerleri belirlemek için dört farklı interpolasyon yöntemi desteklenir:

- **En Yakın Komşu (Nearest Neighbor):** En basit ve en hızlı yöntemdir. Yeni piksel değerini, en yakın orijinal pikselin değerinden alır. Keskin kenarlar ve "aliasing" etkisi oluşturabilir.
- **Bilinear İnterpolasyon:** Dört komşu pikselin değerlerini kullanarak doğrusal bir ağırlıklı ortalama hesaplar. Daha yumuşak geçişler sağlar ve orta kalitede sonuçlar verir.
- **Bicubic İnterpolasyon:** Daha geniş bir piksel komşuluğunu (genellikle 4x4) kullanarak daha karmaşık bir interpolasyon uygular. Yüksek kaliteli, daha yumuşak ve doğal görünümlü sonuçlar üretir, ancak hesaplama maliyeti daha yüksektir.
- **Ortalama (Average):** 2x2 komşu pikselin ortalamasını alarak basit bir bulanıklaştırma etkisi yaratır.

Parametre Yapısı:

- **Ölçek faktörü kontrolü:** Boyutlandırma işlemleri için pozitif bir sayı.
- **Zoom merkezi koordinatları:** 0-1 aralığında normalize edilmiş (x, y) koordinatları, zoom'un uygulanacağı merkezi belirler.
- **Döndürme açısı:** Derece cinsinden döndürme açısı.

5.2 S-Curve Kontrast Güçlendirme

Bu işlem, görüntülerin kontrastını artırmak için sigmoidal (S-şekilli) transfer fonksiyonları kullanır. Bu tür fonksiyonlar, orta tonları vurgularken çok parlak veya çok karanlık alanlardaki ayrıntıları korumaya yardımcı olur.

Fonksiyon Türleri:

- **Standart Sigmoid:** $f(x)=1+e^{-a(x-0.5)}$ formülüne dayalıdır ve orta gri seviye etrafında simetrik bir kontrast artışı sağlar.
- **Yatay Kaydırılmış Sigmoid:** $f(x)=1+e^{-a(x-b)}$ formülü ile orta noktanın (b) ayarlanmasına izin verir, böylece kontrast artışı belirli bir parlaklık aralığına kaydırılabilir.
- **Eğimli Sigmoid (slope_based):** Temelde standart sigmoid ile aynıdır, ancak parametre a (eğim) ile kontrastın şiddeti kontrol edilir.
- **Özel Fonksiyon (custom_tanh):** $21 (\tanh(a(x-0.5)))+1$ formülüne dayalıdır. tanh fonksiyonu da sigmoidal bir şekle sahip olup kontrastı ayarlamak için kullanılabilir.

Kontrol Parametreleri:

- **Eğim parametresi (slope_a):** Fonksiyonun eğimini kontrol eder, bu da kontrast güçlendirmesinin şiddetini belirler. Daha büyük değerler daha keskin kontrast artışı sağlar.
- **Kayma parametresi (shift_b):** Sigmoid fonksiyonunun yatay pozisyonunu ayarlar, böylece belirli parlaklık seviyelerine odaklanılabilir.

5.3 Hough Transform

Hough Transform, bir görüntüdeki belirli geometrik şekilleri (genellikle çizgiler veya daireler) tespit etmek için kullanılan güçlü bir özelliktir.

Tespit Türleri:

- **Çizgi Tespiti (lines):** Görüntüdeki doğrusal yapıları algılar. OpenCV'nin HoughLinesP fonksiyonu ile, çizgiler uç noktaları (x1, y1, x2, y2) olarak temsil edilir.
- **Göz Tespiti (eyes):** Özel bir yüz özelliği tespiti uygulamasıdır. OpenCV'nin Haar Cascade sınıflandırıcısı (cv2.CascadeClassifier) kullanılarak yüzlerdeki gözler algılanır.

Parametreler:

- **Threshold:** Tespit hassasiyeti. Hough dönüşümü alanında yeterli oy toplayan noktaların çizgi olarak kabul edilmesi için gerekli minimum oy sayısı.
- **Min Line Length:** Tespit edilecek çizgilerin minimum uzunluğu (piksel cinsinden).
- **Max Line Gap:** Aynı çizgi üzerinde olarak kabul edilebilecek maksimum boşluk (piksel cinsinden).

5.4 Deblurring (Motion Blur Düzeltmesi)

Bu işlem, hareket bulanıklığı nedeniyle kalitesi düşmüş görüntüleri eski haline getirmeyi amaçlar.

Yaklaşım:

Uygulama, hareket bulanıklığını gidermek için **Wiener dekonvolüsyon** tabanlı bir yaklaşım kullanır. Bu, bulanıklık çekirdeğini (**Point Spread Function – PSF**) kullanarak görüntünün Fourier Uzayında filtrelenmesi prensibine dayanır. Bu yaklaşım, görüntüdeki gürültüyü de dikkate alarak daha iyi sonuçlar elde etmeyi hedefler.

Parametreler:

- **Kernel Boyutu:** Bulanıklık çekirdeğinin boyutu. Pozitif tek sayı olmalıdır. Daha büyük çekirdekler, daha şiddetli veya daha uzun mesafeli bulanıklığı gidermek için kullanılır.
- **Açı:** Hareketin yönü (derece cinsinden). Doğru açının belirlenmesi, deblurring işleminin etkinliği için kritik öneme sahiptir.

create_motion_blur_kernel Fonksiyonu:

Bu fonksiyon, belirtilen **size** ve **angle_degrees** değerlerine göre bir hareket bulanıklığı **PSF'si (Point Spread Function)** oluşturur. Kernel, temelde belirli bir açıda ve uzunlukta bir çizgidir. Bu çizgi, görüntüdeki her pikselin hareket sırasında nasıl yayıldığını temsil eder. Kernel, toplamı 1 olacak şekilde normalleştirilir.

apply_deblur_operation Fonksiyonu:

1. **Giriş Kontrolü:** Görüntünün yüklü olup olmadığı ve kernel boyutunun geçerli (pozitif tek sayı) olup olmadığı kontrol edilir.
2. **PSF Oluşturma:** `create_motion_blur_kernel` fonksiyonu ile hareket bulanıklığı çekirdeği oluşturulur.
3. **Görüntüyü Kayan Noktalı Hale Çevirme:** Görüntü, FFT işlemleri için 0-1 aralığında normalize edilmiş `float32` tipine dönüştürülür.
4. **Kanal Bazında İşleme:** Deblurring, her RGB kanalı için ayrı ayrı yapılır.
 - **FFT Dönüşümü:** Her bir kanalın ve PSF'nin 2D FFT'si (`np.fft.fft2`) hesaplanır. PSF, görüntünün boyutuyla eşleşecek şekilde sıfırlarla doldurulur.
 - **Basitleştirilmiş Wiener Filtreleme:** Frekans uzayında, bulanıklaştırılmış görüntünün FFT'si, Wiener filtresi formülünün basitleştirilmiş bir versiyonuyla çarpılır: $WienerFilterFFT = |PSFFFT|^2 + K \cdot conj(PSFFFT)$ Burada K (noise factor), gürültü-sinyal güç oranının bir tahmini olup, kullanıcının ayarlaması gereken bir parametredir. Daha küçük K değerleri daha agresif deblurring, daha büyük değerler ise gürültüye karşı daha fazla direnç sağlar.
 - **Ters FFT:** Filtrelenmiş FFT'nin ters 2D FFT'si (`np.fft.ifft2`) alınarak deblurred kanal elde edilir. Sonuçların mutlak değeri alınır.
5. **Sonuçları Birleştirme ve Normalleştirme:** İşlenmiş kanallar birleştirilerek `uint8` formatında 0-255 aralığına geri dönüştürülür ve `convert_cv_to_txt` ile özel TXT formatına çevrilir.

5.5 Nesne Tespiti

Bu bölüm, belirli renk aralıklarındaki nesneleri tespit etmeye odaklanır, özellikle koyu yeşil bölgelerin tespiti için yapılandırılmıştır.

Yöntem:

- **HSV renk uzayında koyu yeşil bölge tespiti:** RGB renk uzayı yerine **HSV (Hue, Saturation, Value)** renk uzayı kullanılır, çünkü renkleri parlaklık ve doygunluktan bağımsız olarak ayırmak daha kolaydır. Belirli bir renk aralığı (koyu yeşil için alt ve üst HSV sınırları) tanımlanarak görüntüde bu renge sahip pikseller maskelenir.
- **Morfolojik Operasyonlar:** Maske üzerinde açıklık (`MORPH_OPEN`) ve kapanıklık (`MORPH_CLOSE`) gibi morfolojik operasyonlar uygulanarak gürültü temizlenir ve nesne şekilleri daha düzgün hale getirilir.
- **Kontur Bulma:** Temizlenmiş maske üzerinde `cv2.findContours` kullanılarak nesne konturları bulunur.
- **Özellik Çıkarma:** Bulunan her bir nesne (kontur) için çeşitli geometrik ve piksel tabanlı özellikler çıkarılır.

Parametreler:

- **HSV Alt Sınır:** (H, S, V) değerleri, tespit edilecek rengin alt sınırını tanımlar.
- **HSV Üst Sınır:** (H, S, V) değerleri, tespit edilecek rengin üst sınırını tanımlar.

Çıktı ve Özellikler:

- Tespit edilen nesnelerin özellikleri (`object_features_data` listesine eklenir):
 - **Merkez:** Nesnenin ağırlık merkezi (momentler kullanılarak).
 - **Uzunluk ve Genişlik:** `minAreaRect` ile elde edilen dönmüş dikdörtgenin kenar uzunlukları, nesnenin yönünden bağımsız boyutlarını verir.
 - **Köşegen:** `minAreaRect`'in köşegen uzunluğu.
 - **Ortalama Yoğunluk (Mean Intensity):** Nesnenin üzerindeki piksel yoğunluklarının ortalaması.
 - **Medyan Yoğunluk (Median Intensity):** Nesnenin üzerindeki piksel yoğunluklarının medyan değeri.
 - **Enerji:** Görüntü histogramının enerji değeri, doku düzenini gösterir.
 - **Entropi:** Görüntü histogramının entropi değeri, doku karmaşıklığını gösterir.
- Bu özellikler, `result_cv` görüntüsü üzerinde konturlar, merkez noktaları ve dönmüş sınırlayıcı kutular çizilerek görselleştirilir.
- Tespit tamamlandığında, bulunan nesne sayısı kullanıcıya bilgi mesajı ile bildirilir.

5.6 Nesne Özelliklerini Excel'e Kaydetme

- **Fonksiyon:** `save_to_excel()`
- Bu fonksiyon, `apply_object_detection` işlemi sonucunda elde edilen `object_features_data` listesini bir **Pandas DataFrame**'e dönüştürür ve kullanıcı tarafından belirtilen bir yola `.xlsx` formatında bir Excel dosyası olarak kaydeder.
- Kullanıcıya kaydetme konumu için bir dosya seçim diyalogu (`filedialog.asksaveasfilename`) sunulur.
- Kaydetme işlemi başarılı olduğunda veya bir hata oluştuğunda ilgili `messagebox` ile geri bildirim sağlanır.

6. Görüntü Görselleştirme Sistemi

6.1 Özel TXT Format Desteği

Uygulama, görüntüleri bir **özel TXT formatında** saklama ve işleme yeteneğine sahiptir.

- **Format:** `[width][height]` şeklinde başlangıç satırı ve ardından her piksel için `[R][G][B]` değerlerinin sıralı bir dizisi şeklinde yapılandırılmıştır.
- **Avantaj:** Ham piksel verilerine doğrudan erişim sağlar, bu da algoritma geliştirme ve test amaçları için oldukça kullanışlıdır. Özellikle özel interpolasyon algoritmaları ve geometrik dönüşümler için piksel seviyesinde kontrol sağlar.

6.2 Görüntü Gösterim Optimizasyonu

- **Otomatik Boyutlandırma:** Büyük görüntüler, GUI'deki görüntüleme alanına sığacak şekilde otomatik olarak küçültülür.
- **Oransal Koruma:** Görüntülerin orijinal en-boy oranı (`aspect ratio`) korunarak thumbnail'ler oluşturulur, bu da görüntülerin bozulmadan gösterilmesini sağlar.
- **Bellek Yönetimi:** PIL Image referansları uygun şekilde yönetilir, gereksiz bellek kullanımının önüne geçilir.

7. Hata Yönetimi ve Kullanıcı Deneyimi

7.1 Exception Handling

Uygulama, sağlam bir hata yönetimi (`exception handling`) mekanizması ile donatılmıştır:

- **Görüntü yükleme hataları:** Geçersiz dosya yolları veya bozuk dosyalar için `messagebox` uyarıları gösterilir.
- **Geçersiz parametre girişleri:** Kullanıcıdan beklenen sayısal değerlerin doğru formatta girilip girilmediği kontrol edilir (örn. pozitif ölçek faktörü, geçerli açılar). Geçersiz girişlerde hata mesajları görüntülenir.
- **Boş görüntü verisi durumları:** İşlem uygulanmadan önce gerekli görüntü verisinin (OpenCV formatı veya özel TXT formatı) mevcut olup olmadığı kontrol edilir.
- **Spesifik Hata Yakalama:** HSV girişleri için `ValueError` gibi spesifik hatalar yakalanarak daha bilgilendirici mesajlar sunulur.

7.2 Kullanıcı Geri Bildirimi

- **İşlem başarı/başarısızlık durumları:** İşlemlerin başarıyla tamamlanması veya bir hatayla karşılaşılması durumunda kullanıcıya görsel geri bildirim sağlanır (örn. `messagebox` uyarıları veya durum çubuğu mesajları).
- **Parametre değişikliklerinin dinamik güncellenmesi:** GUT'deki parametre giriş alanları, kullanıcıların değerleri kolayca değiştirmesine ve işlemleri yeniden uygulamasına olanak tanır.
- **Sezgisel kontrol paneli düzeni:** Modüler tasarım, kullanıcıların aradıkları işlemleri ve parametreleri kolayca bulmalarını sağlar.

8. Sonuç ve Değerlendirme

Bu kod bölümü, kapsamlı dosya I/O sistemi ve gelişmiş geometrik dönüşüm algoritmalarını içermektedir. İnterpolasyon yöntemlerinin çeşitliliği ve sağlam hata yönetimi ile birlikte, proje güçlü bir temel üzerine inşa edilmiştir. Son eklenen Deblurring ve Nesne Tespiti/Özellik Çıkarma fonksiyonları, uygulamanın yeteneklerini önemli ölçüde artırmaktadır.

Güçlü Yönler:

- **Çoklu format desteği:** OpenCV, PIL ve özel TXT formatları arasında sorunsuz geçiş yeteneği, farklı işlem türleri için esneklik sağlar.
- **Dört farklı interpolasyon yöntemi:** Kullanıcılara, hız ve kalite arasında seçim yapma imkanı sunar.
- **Robust hata yönetimi:** Görüntü yükleme, parametre girişleri ve işlem sırasında oluşabilecek hatalara karşı dayanıklıdır.
- **Matematiksel olarak doğru algoritma implementasyonları:** Geometrik dönüşümler, interpolasyonlar ve bulanıklık giderme işlemleri, görüntü işleme prensiplerine uygun şekilde uygulanmıştır.
- **Kapsamlı nesne tespiti ve özellik çıkarma:** Nesneleri algılamak ve onlar hakkında detaylı sayısal bilgiler (merkez, boyut, yoğunluk, entropi) çıkarmak, ileri düzey analizlere olanak tanır.

- **Excel'e veri aktarımı:** Tespit edilen nesne özelliklerinin raporlanabilir bir formatta (Excel) dışa aktarılabilmesi, uygulama çıktılarının pratik kullanımını artırır.

Teknik Başarılar:

- **Bicubic interpolasyon ile yüksek kalite sonuçlar:** Özellikle büyütme ve döndürme işlemlerinde görsel olarak üstün sonuçlar elde edilmesini sağlar.
- **Memory-efficient veri yapısı tasarımı:** Bellek kullanımını optimize etmek için uygun veri yapıları ve referans yönetimi kullanılmıştır.
- **Modüler interpolasyon sistemi:** Yeni interpolasyon yöntemlerinin kolayca eklenebileceği esnek bir yapı sunar.
- **Güvenli boundary handling:** Piksel erişiminde sınır dışı durumlar için sağlam koruma mekanizmaları bulunur.
- **Wiener dekonvolüsyonunun FFT tabanlı uygulanması:** Bulanıklık giderme için karmaşık ancak etkili bir yöntemin temel prensipleri entegre edilmiştir.
- **Detaylı nesne özellik çıkarma:** Kontur tabanlı analizler ile nesnelerin sadece tespiti değil, aynı zamanda nicel özelliklerinin de elde edilmesi sağlanmıştır.