****

GÖRÜNTÜ İŞLEME EDİTÖRÜ

Ilyas Mammadov, Dilara Nur Yavuz

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[ilyas.mammadov.96@gmail.com](mailto:ilyas.mammadov.96@gmail.com), [dilaranuryavuz@gmail.com](mailto:dilaranuryavuz@gmail.com)

# Özet

# Projenin konusu basit bir görüntü işleme editörü yazılımıdır. Bu yazılım kısaca şunları yapabilmelidir: Bir RGB görüntüyü gri tonlu görüntüye çevirme, negatifini alma, görüntüyü sağa ve sola aynalama, saat yönünde ve saatin tersi yönünde 90° döndürme, görüntünün gri ve renkli histogramını alma, Red-Green-Blue kanallarını ayrı-ayrı göstermelidir.

Görüntüyü programa aktarmak için OpenFileDialog sınıfından faydalandım. Daha sonra bu görüntüyü bir Bitmap nesnesine atayarak, matrissel işlemler yardımı ile görüntü üzerinde değişiklikler yapılmasını sağladık.

Proje çalışması sırasında herhangibir hazır kaynak kullanılmamıştır. Kullanılması pilanlanan materyel hakkında araştırma yapılarak projenin özgün tutulmasına özen gösterilmiştir.

# Giriş

Proje konusu ile ilgili daha önce yapılan ve dünyaca tanınan Adobe Photoshop, Paint.NET, Picasa, Corel Paint Shop Pro Photo gibi projeler mevcuttur. Biz projemizin kalite ve testini yaparken çıktıları Photoshop programı ile kıyaslayarak gözlemlenen eksiklikleri gidermeye çalıştık.

Bu projenin amacı basit bir görüntü işleme yazılımı yazmak. Bu yazılım şunları yapabilmeli: Bir RGB görüntüyü gri tonlu görüntüye çevirme, negatifini alma, görüntüyü sağa ve sola aynalama, saat yönünde ve saatin tersi yönünde 90° döndürme, görüntünün gri ve renkli histogramını alma, Red-Green-Blue kanallarını ayrı ayrı göstermelidir. Proje sonucu elde edilen çıktılar 100% doğru yanıt vernekle beraber yokarıda belirtilen tüm amaçlar gerçeklenmiştir.

# Temel bilgiler

Projenin gerçeklenmesinde, yazılan kodlar C# dili ile yazılmıştır. IDE olarak Visual Studio kullanılmış, arayüz geliştirmesinde ise internetten temin edilen bazı hazır görseller ve bir vektörel çizim harikası olan Corel Draw programından faydalanılmıştır. Bunun haricinde gerekli bilgiyi elde edebilmek ve gerekli yazılımları indirebilmek adına internetten faydalanılmıştır.

# Diğer bölümler

* **Bilgi toplama aşaması**

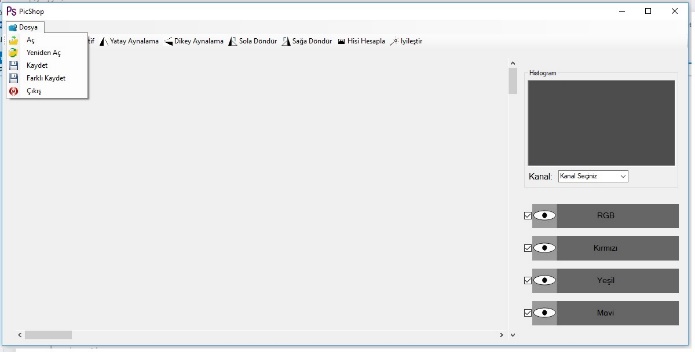
Bu aşamada ilk önce kullanılabilecek teknolojiler hakkında bilgi toplanarak hangi aşamada hangi teknolojinin kullanılacağına karar verilmiştir. Örneğin projenin kodlanacağı programlama dilini seçerken C# programlama dilinin zaman konusunda ve arayüz konusunda daha kolay ve hızlı olacağından programın bu dilde yazılması öngörülmüştür. Dosyalama işlemlerinin hangi kütüphane üzerinden yapılacağı, setup dosyası oluşturma işleminin nasıl yapılacağı gibi konular bu aşamada kararlaştırılmıştır.

* **Arayüz tasarlanması**

Arayüzü tasarlarken genel Windows Form Application uygulamalarının tasarımı göz önünde bulundurularak aynı zamanda görüntü işleme konusunda namı değer Photoshop programının kullanıcı arayüzünden de ilham alınmıştır. Alttaki resimde görüldüğü gibi form penceresinin en üstüne yatay olarak bir MenuStrip yerleştirilmiş ve dosyalama işlemleri, görünüm ayarları, program ayarları ve yardım menüsü işlemleri buradan yapılması amaçlanmıştır.

Hemen bu menünün altına gene yatay bir şekilde konumlandırılmış ToolStrip(Araç Şeridi) yerleştirilmiştir. Bu şeritte ise Gri Yap, Negatif, Yatay Aynala, Dikey Aynala, Sola Döndür, Sağa Döndür, Histogram Hesapla, Görüntüyü İyileştir gibi görüntü işleme araçları yer almaktadır.

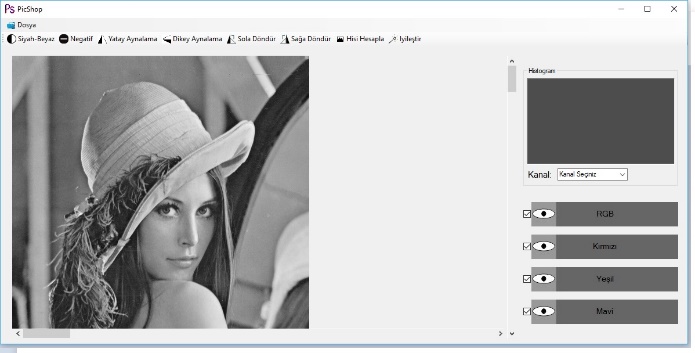
Pencerenin sağ üst köşesinde görüntü ile ilgili hisogram bilgilerini sunacak bir histogram alanı mevcut. Bunun hemen altında ise görüntünün renk kanallarının kontrolünü sağlayacak CheckBox’lar yer almaktadır. Geri kalan kısımda ise üzerinde çalıştığımız görüntü ile ilgili değişiklikleri görebileceğimiz bir PictureBox nesnesi ve bu PictureBox’un sağında ve alt kısmında iki adet ScrollBar yer almakta.



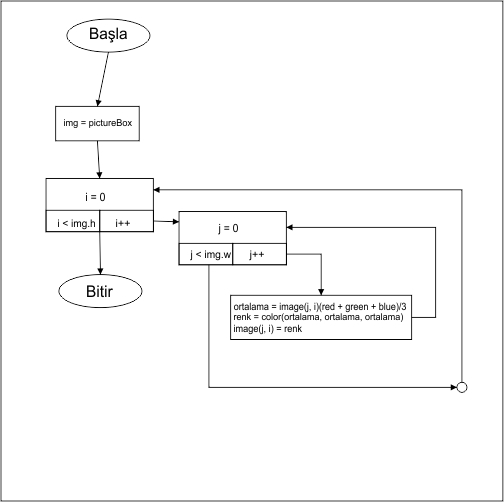
Şekil 1: Açılış ekranı görüntüsü.

* **Algoritma tasarımı**
  + *griyap() fonksiyonu*

griyap() fonksiyonu RGB görüntüyü alarak gri tonlu görüntüye dönüştüren bir fonksiyondur algoritması aşağıdaki gibidir.



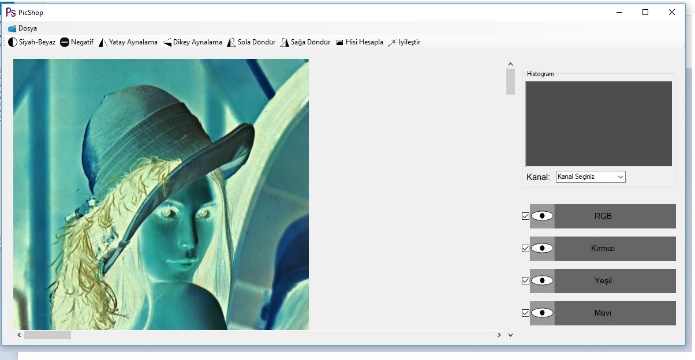
Şekil 2: Gri tonlu görüntüye çevrilmiş resim.



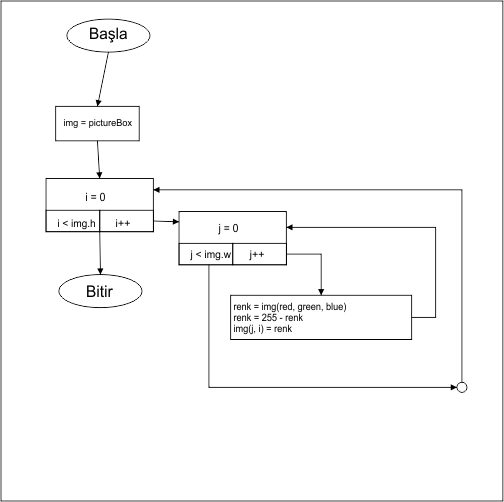
Şekil 3: griyap() fonksiyonunun akış şeması.

* + *negatif() fonksiyonu*

negatif() fonksiyonu görüntünün negatifini alarak ekrana yansır. Algoritması aşağıdaki gibidir.



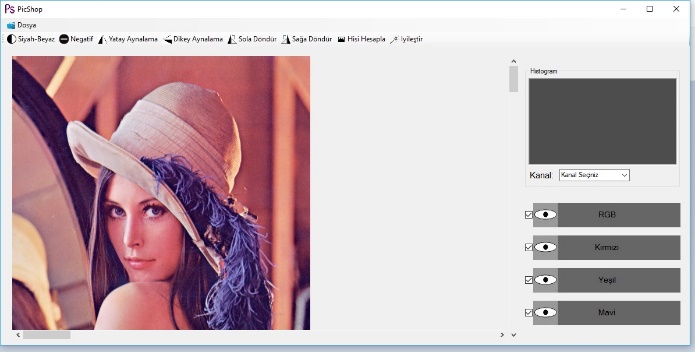
Şekil 4: negatif() fonksiyonunun örnek çıktısı.



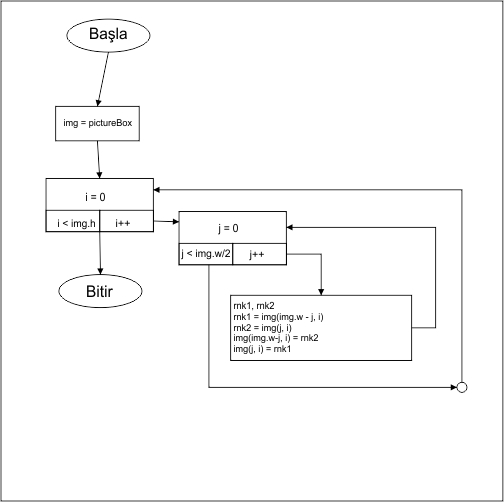
Şekil 5: negatif() fonksiyonunun akış şeması.

* + *yatayaynala() fonksiyonu*

yatayaynala() fonksiyonu görüntüyü yatay olarak aynalanmış hale getirir. Algoritması ve örnek çıktısı aşağıdaki resimde görülmektedir.



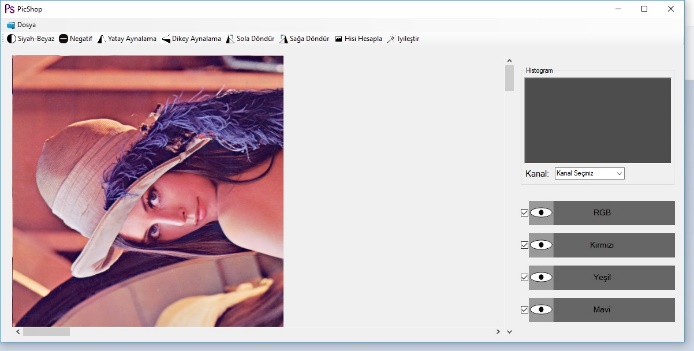
Şekil 6: yatayaynala() fonksiyonunun örnek çıktısı.



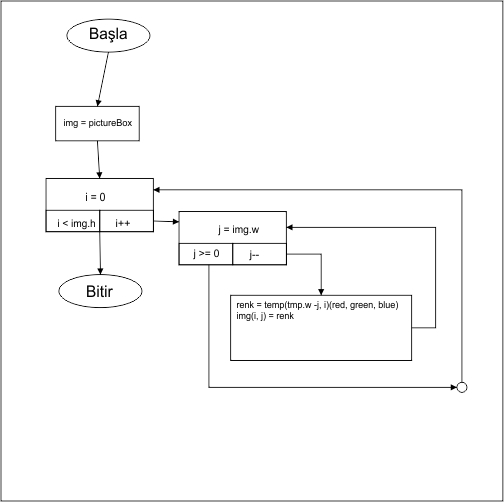
Şekil 7: yatayaynala() fonksiyonunun akış şeması.

* + *soladondur() ve sagadondur() fonksiyonu*

soladondur()/sagadondur() fonksiyonu görüntüyü saatin tersi yönünde/saat yönünde 90 derecelik açı ile döndürür. Algoritması ve örnek çıktısı aşağıdaki resimdeki gibidir.



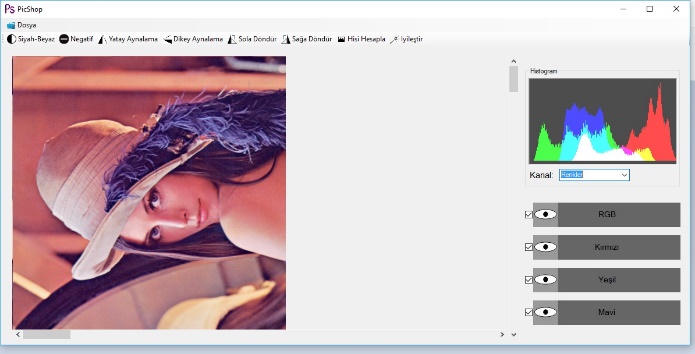
Şekil 8: soladondur() fonksiyonunun örnek çıktısı.



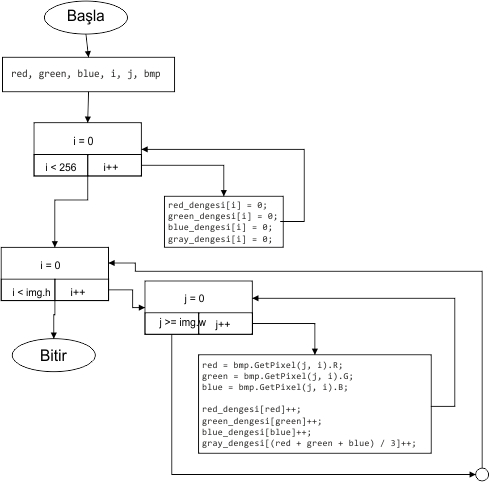
Şekil 9: soladondur() fonksiyonunun akış şeması

* + *hisihesapla() ve histogram() fonksiyonu*

hisihesapla() fonksiyonu görüntüdeki her rengin her tonundan görüntüe kaç adet bulunduğunu hesaplar ve bu sonuçları histogram fonksiyonunda kullanmak üzere bir dizide tutar. histogram() fonksiyonu ise bu verileri kullanarak gerekli matematiksel ve mantıksal olarak histogram haline dönüştürecek şekilde işledikten sonra 256x150 px boyutundaki Bitmap nesnesinde bu yeni verilere uygun olarak hektogramın işler ve histogram penceresine basarak görsel bilgi haline getirir. Örnek çıktı ve kodun algoritması ile ilgili akış şemasını aşağıdaki resimlerde görebilirsiniz.



Şekil 10: Hesaplanmış histogram görüntüsü.



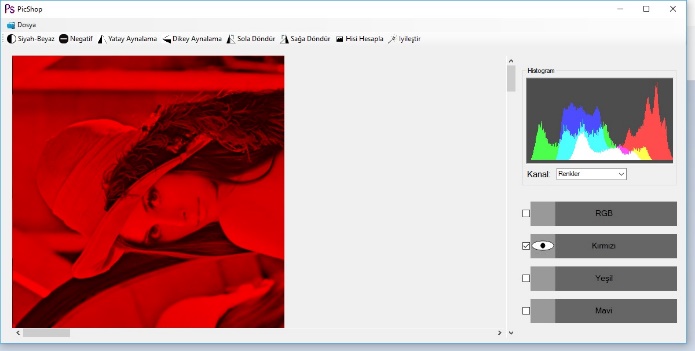
Şekil 11: hisihesapla() fonksiyonunun akış şemaı.

* + *iyilestir() fonksiyonu*

Bu fonksiyon görüntünün her renk kanalı için renk tonu aralıklarını hesaplayarak, eğer renk bileşenlerinin ton aralıkları 256’dan küçük ise, bazı matematiksel ve mantıksal formülizasyonlardan sonra renk tonu aralığını maksimuma çıkartarak renklerin bir birinden daha iyi ayrıştırımasını sağlayarak, daha kaliteli görüntü oluşturur.

* + *renkkanali() fonksiyonu*

Bu fonksiyon çalıştırldığında ilgili CheckBox’ların durumunu kontrol ederek uygun gelen kod bloğunu çalıştırır ve ilgili işlemler sonucu CheckBox’lar vasıtası ile seçilen renk bileşenleri ekranda gösterilir. Bu fonksiyonun da örnek çıktısını ve akış şemasını aşağıda görebilirsiniz.



Şekil 12: Renk kanalı örnek çıktı.

* **Setup oluşturma**

Setup oluşturma Visual Studio’nun bir eklentisi olan setup oluşturucu bir program yardımı ile yapılmıştır. Programda belirtilen gerekli adımları ve düzenlemeleri ayarladıktan sonra program setup’u oluşturarak bir klasör haline getiriyor ve kurulum dosyalarını bu klasörün içerisine yerleştiriyor.

# Sonuçlar

Bu projede bizden istenen basit bir görüntü işleme arayüzü geliştirmekti. Proje tanımında yer alan bütün maddeler başarılı bir şekilde gerçekleştirilmiştir ve elde ettiğimiz sonuçlar tamamen amaca uygundur. Proje gerçekleştirme sürecinde karşılaşılan belli başlı problemler:

\*Görüntüyü ölçeklendirirken pencere boyutlarını aşan görüntü boyutuna karşın görüntüyü çalışma ekranında kaydırarak istediğimiz bölümü çalışma ekranında gösterememe. Bu problemi görüntüyü pencerede göstermeye yarayan PictureBox’ı bir panel içerisine yerleştirerek görüntünün orijinal boyutunu değişmek yerine PictureBox boyutunu değiştirerek çözdük ve böylece kaydırma problemi de ortadan kalkmış oldu.

# Kaynakça

Her bir kaynak aşağıda verildiği gibi numaralandırılmalıdır. Her bir kaynak mutlaka rapor içinde gösterilmelidir. Salt kaynak listesi sıralamayınız. Rapor içinde verdiğiniz kaynaktan nerede yararlandı iseniz hemen yanında şu şekilde atfediniz [1] gibi.

1. Yazar soyadı, Adının ilk harfi ve varsa aynı şekilde diğer yazarlar, “Çalışmanın adı”, Çalışmanın yayınlandığı dergi (veya internet sayfası gibi), yayın yılı.
2. Smith, J. O. and Abel, J. S., ``Bark and ERB Bilinear Transforms'', IEEE Trans. Speech and Audio Proc., 7(6):697-708, 1999.