FOTOĞRAF DÜZENLEME PROJE RAPORU

28 MAYIS 2021

Abdulsamet Alkan

Ahmet Enes Zor

İÇİNDEKİLER

1. GİRİŞ

21. yüzyılın değişen yaşam pratikleri hızla gelişen teknolojiyle entegre hale gelmeye başlamış ve boyutları bir kıyafet cebine sığdırılabilecek kadar küçültülmüş bilgisayarlarla istenilen anların dijital dünyada "ölümsüzleştirilmesi" mümkün hale gelmiştir. İşlem kapasitesinden ya da estetik görünümünden taviz vermeden günlük yaşama adapte edilmeye çalışılan bilgisayarlar, yeni dünya insanının yaşam biçimine uygun hale getirilmesi için dönüştürülmeye hazır olarak tasarlanmaktadır. Bunun da ötesinde "ölümsüzleştirilen" bu fotoğrafların yayılmasını sağlama görevi ise küreselleşmenin katalizörlerinden birisi haline gelen sosyal medya ağlarına düşmektedir. Bu ağlar aracılığıyla dünyanın neresinde olduğu fark etmeksizin gerekli cihazı olan her insana ulaşabilmek vadedilmiş ve akabinde başarıya da ulaştırılmıştır.

Kişisel tatminin yanında fotoğrafların kazandığı sosyal etki hayatın yadsınamaz bir gerçekliği olmuştur. Fiziki uzaklıklara meydan okuyan fotoğraflar dünyanın en azından dijital erişilebilirlik manasında "düz" hale geldiğinin bir kanıtı mahiyetini taşımaktadır. Birbirini tanımayan ama birbirlerinin hayatlarını merak eden ve başkalarıyla paylaşma isteği duyan insanların arasındaki yapay köprüyü oluşturan fotoğraflar insan hayatının vazgeçilmez bir parçası olmuşlardır.

Bütün bu dönüşüme rağmen çekilen fotoğraflar ışık şiddeti, çekim açısı gibi doğrudan ortam değişkenlerinden ya da dolaylı birçok değişkenden ötürü her zaman istenilen güzellikte ve kalitede olmayabiliyor. Bu noktada, ölümsüzleştirilmesi vadedilen görüntünün insan gözüyle edinilen kaliteden uzaklaşılması bu eyleme karşı duyulan tatminin azalmasına sebep olmaktadır. Dahası bu fotoğraflarda orada olmasından memnun olunmayan nesnelerin yahut da kişilerin varlığı fotoğrafın kalitesini etkileyen unsurlardan olabilmektedir.

Günlük hayatta aynı fotoğrafın tekrar çekilmesi en basit yolmuş gibi görünse de fotoğrafın çekildiği anın biricikliğinin bu yöntemle kaybolma riski taşıdığı bilinmektedir. Nitekim bazı "anların" yeniden çekilebilmesinin olanağı dahi bulunmamaktadır. Sıralanan sorunların çözümü ise, çekilen fotoğrafların gerekli düzenlemelerden geçirilerek istenmeyen unsurlardan arındırılmasını sağlayan uygulamalarda aranmaktadır. İnsanın merkeze alındığı yaklaşımların revaçta olduğu bu dönemde, arzu edilen kalitenin daha hızlı, daha kolay ve daha az maliyetli bir şekilde elde edilerek yeniden sunulmasının imkânının tanınması ise oldukça önemlidir. Daha önce yapılmış yazılımların bu noktalardaki eksiklikleri göz önünde tutularak bir yazılım tasarımı planlanmıştır.

Bu proje, durmak bilmeyen teknolojik dünyada herkese ücretsiz ve açık kaynaklı bir yazılım hizmeti sağlayarak ele alınan soruna gerekli ve basit bir çözüm hedeflemektedir. Herkesin bildiği gibi birçok yazılım ya tamamıyla ücretli ya da yazılımın verdiği hizmetin kısmi özellikleri ücretsiz gelişmiş özellikleri ise bir ücrete tabi tutuluyor. Özellikle de fotoğraf düzenleme uygulamalarında bir hayli karşılaşılan bu durum teknolojik imkânlara maddi nedenlerden ötürü tam anlamıyla erişim sahibi olamayan insanları küçük bir alanda kısıtlıyor. Özgür yazılıma inanan bireyler olarak bu kısıtlamaları yıkmak bir gereklilik haline gelmiştir. Bu durumda isteyen herkesin katkıda bulunabileceği böylece gelişime açık ve kullanımının ücretsiz olması öngörülen bu projeyi tanıtarak erişim imkânı sunmanın gerekliliği ortadır.

Proje sayesinde kullanıcılar siyah beyaz formattaki (pgm formatındaki) fotoğraflara birtakım efektler uygulayabilir veya fotoğraftan herhangi bir nesneyi silebilirler. Buradaki kritik nokta ise nesne silmek için bir fotoğrafın birden çok farklı anlarda çekilmiş kareleri gerekmektedir.

Programı çalıştırabilmek için bir bilgisayar ortamında konsol kullanımı gerekmektedir. Program, konsol (komut istemi) üzerinden gerekli girdiler verilerek çalıştırılır ve istenilen çıktılar alınır. *Adobe* gibi emsallerine kıyasla uygulamaya dair birçok teknik özelliği öğrenme çabasına gerek kalmadan sadece istenilen efekt ve fotoğraf programa verilerek saniyeler içinde sonuç alınabilir. Bu proje henüz gelişim aşamasında olduğu için rakipleri kadar -örneğin *Adobe*- kadar gelişmiş ve karmaşık bir yapıya sahip değildir. Öte yandan konsol üzerinden kullanıldığı için proje için geliştirilmiş bir grafik arabirimi bulunmamaktadır.

Proje %100 C programlama dili ile geliştirilmiştir. Bilgisayar ile daha hızlı bir şekilde iletişime geçmesi ve tepkiselliği üst seviyelere çekmesiyle bilinen C programlama dili sayesinde kullanıcılara daha hızlı bir program deneyimi sunulmaktadır.

2. PROJE MİMARİSİ

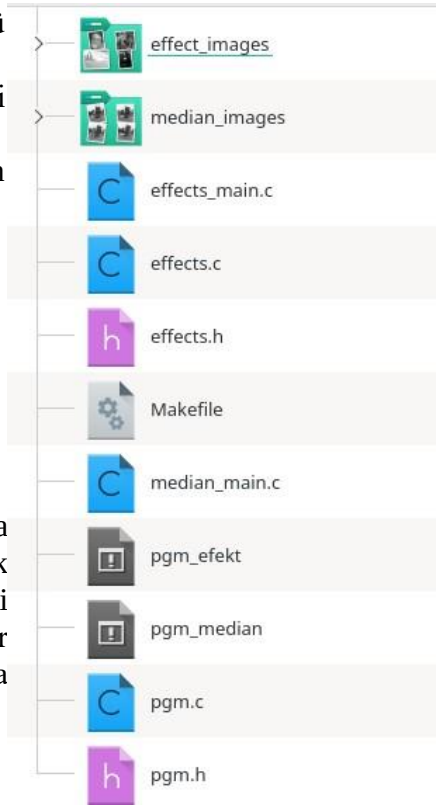
2.1 Teknoloji ve Destek

Proje yapımında bilgisayar teknolojilerinde yaygın bir hal almış C programlama dilinden ve dolayısıyla bu dile ait kütüphane ve derleme araçlarını beraberinde getiren *GCC* 10.2.0 sürümünden yararlanılmıştır. Geliştiriciler tarafından oluşturulan "pgm.h" ve "effects.h" kütüphaneleri, öte yandan GCC tarafından sağlanan "stdio.h", "stdlib.h", "string.h", "assert.h", "time.h" ve "limits.h" kütüphaneleri proje mimarisinde kullanılmış, Makefile ile projenin derlenmesi son derece kolay bir hale getirilmiştir. Geliştirme esnasında *Visual Studio Code* ve *Eclipse* yazılım geliştirme ortamı kullanılmış, Linux işletim sistemi tercih edilmiştir. Program *Manjaro, Pop!_OS, Ubuntu, Linux Mint* ve *Garuda* gibi Arch Linux ve Debian tabanlı dağıtımlarda denenmiş ve Arch-Debian bazlı çoğu Linux dağıtımında çalışacağı teyit edilmiştir.

2.2 Proje İç Yapısı ve Kullanımı

Proje için hazırlanan dosya yapısı yandaki fotoğrafta da görüldüğü üzere 3 ana kısımdan oluşmaktadır:

1. effect_images: Düzenleme yapılacak fotoğrafların yerleştirildiği klasör.
2. median_images: Filtreleme işlemiyle varlığı silinmesi istenen nesneleri içeren fotoğraflar bu klasörde yer alır.
3. Bu 2 klasörün dışında bulunan ve proje kodlarını içeren dosyalar.



*Proje için özel olarak geliştirilen kütüphane kaynak kodları "pgm.c" ve "effects.c" içerisine yazılmıştır.

***Pgm.c*:** İçersinde *pgm_print_header* , *pgm_read* ve *pgm_write* fonksiyonlarını barındırır.

void pgm_print_header(PGMInfo pgm_info): Bu fonksiyon programa verilem pgm formatındaki fotoğrafların imza, en ve boy(çözünürlük bilgilerini) ekrana yazdırır. Fonksiyon parametre olarak *PGMInfo* veri tipinde bir değişken alır. *PGMInfo* "pgm.h" dosyasında tanımlanmış bir veri tipidir. PGM dosyalarının başlık bilgileri gibi fotoğraf hakkında birtakım bilgiler içerir.

PGMInfo pgm_read(const char *filename): Fonksiyona verilen pgm dosyasındaki bilgiler *PGMInfo* tipinde bir değişkene atanır. Böylece fotoğraf üzerinde değişiklik yapılabilir.
int pgm_write(const char *filename, PGMInfo pgm_info): Fonksiyona üzerinde değişiklik yapılmış, oluşturulacak fotoğrafın bilgilerini tutan *PGMInfo* veri tipinde bir değişken verilerek "filename" değişkeniyle verilen dosya ismiyle yeni bir fotoğraf ortaya çıkartılır.

Effects.c: İçersinde effect_random_noise , effect_invert , effect_threshold , effect_contrast ve effect_grey olmak üzere 5 farklı efekt fonksiyonu bulunur. Fotoğraflar bu fonksiyonların yardımıyla düzenlenir.
void effect_random_noise(unsigned char *pixels, int *width, int *height): Fotoğrafı %5 oranında daha gürültülü hale getirir.
void effect_invert(unsigned char *pixels, int *width, int *height): Fotoğraf renklerini tersine çevirerek fotoğrafı zıtlaştırır.
void effect_threshold(unsigned char *pixels, int *width, int *height, int *threshold): Fotoğrafa verilen ve rastgele oluşturulan "threshold" değişkeni eşik değer alınarak 2 renkli siyah ve beyaz bir fotoğraf oluşturulur.
void effect_contrast(unsigned char *pixels, int *width, int *height): Fotoğraf kontrastını arttırmak için kullanılır.
void effect_grey(unsigned char *pixels, int *width, int *height): Fotoğrafın tamamını griye yaklaşan tonlarla düzenler. Sonuç olarak grileştirilmiş bir fotoğraf elde edilir.

*Bu oluşturulan iki kütüphane dosyası "effects_main.c" ve "median_main.c" ana program kodlarında kullanılmıştır.

Effects_main.c: Girdi olarak alınan fotoğraflara istenen efekt burada uygulanır ve çıktı olarak yeni bir fotoğraf oluşturulur. Fotoğraf bilgilerini okumak ve yeni fotoğrafı oluşturmak için "pgm.h" ve gerekli efekt düzenlemelerini uygulamak için "effects.h" fonksiyonu bu dosyaya eklenir.

Median_main.c: Fotoğrafta nesne filtreleyerek silinmesi bu dosya içersinde gerçekleşir. Programa silme işlemi yapılacak fotoğrafın farklı karelerini içeren fotoğraflar verilir. Dosya içinde bulunan *sort_and_get_median* fonksiyonu yardımıyla fotoğrafta filtreme yapılır ve istenen nesne silinir.
unsigned char sort_and_get_median(unsigned char *pixels, int size): Fonksiyona "size" değişkeniyle fotoğraf sayısını, "pixels" değişkeniyle fotoğrafların piksel sayısı verilir. Sonuç olarak fotoğrafta insan gibi hareketli varlıkların oluşturduğu istenmeyen görüntüler için her pikselin medyan ddeğeri bulunur ve bu varlığın oluşturduğu etki ortadan kaldırılır. Diğer bir sözle o varlık silinir.

Program kullanımı efekt uygulaması ve filtreleme için 2 ayrı biçimde kullanılmaktadır:
Efektlerin uygulanması: Bunun için "pgm_efekt" çalıştırılabilir dosyası kullanılır. Çalıştırabilmek için konsol üzerinden girdiler verilerek "enter" tuşuna basılır. Linux için örnek bir kullanım şu şekilde olmalıdır:

./pgm_efekt <invert|binarize|noise|contrast|grey> <PGM image 1> <PGM image 2> ... <PGM image N>

Eğer bir hata kodu alınmaz ise istenen fotoğraf "effect_images" klasöründe istenen efektin adıyla yerini almaktadır.
Filtreleme/Nesne silme: Bu düzenleme "pgm_median" çalıştırılabilir dosyası kullanılarak uygulanır. Düzenleme yapılacak fotoğraflar "median_images" klasörüne yerleştirildikten sonra tek yapılması gereken konsol üzerinden "pgm_median" dosyasını çalıştırmaktır. Örnek bir kullanım şu şekilde olmalıdır:

./pgm_median

Proje, dosyaları derlenmiş bir şekilde kullanıma hazırdır. Fakat tekrar derlenmek istenirse Linux işletim sistemlerinde GCC derleyicisi sisteme yüklenmelidir. Daha sonrasında proje dosyalarının bulunduğu dizinde konsol penceresini açarak <*make*> komutu çalıştırmak yeterlidir.

2.3 Projede İzlenilen Yol Haritası ve Proje Süresi

Proje 4 gün sürmüş, 2 kişi tarafından çevrimiçi iletişim programları aracılığıyla hazırlanmıştır. Her projede olduğu gibi ilk olarak ele alınan sorun ve getirmek istenilen çözüm doğrultusunda gerekli araçlar ve çözüm yolları belirlendi. Daha sonrasında buna uygun bir iskelet kod ve dosya yapısı oluşturularak yapılacak işler küçük adımlara bölündü. En sonda ise bu küçük adımlar sırasıyla tamamlanarak bir bütün oluşturuldu.

2.4 Karşılaşılan Problemler

Proje sonuna yaklaşıldığında, yazılan tüm kodlar derlenme aşamasına geldiğinde artık yapılan ama farkedilmeyen hatalarla karşılaşılır. Bu projede ise şu hatalar alınmış ve çözüme kavuşturulmuştur:

Sysmalloc: Assertion

Fotoğraf bilgilerini depolayan PGMnfo tipindeki değişkende piksellerin atandığı dizi için malloc ile bellekte gerekli miktarda yer açılmasına rağmen bazı fotoğraflarda bu hata alınmıştır. Ayrılan bellek miktarı 1 byte(bayt) arttırılarak çözümü sağlanmıştır.

Başlık Yorumunu Okumak

Fotoğrafların başlık bilgisinde yer alan yorum satırlarını doğrudan PGMInfo tipindeki değişkenlere atamak hem yorum satırında bulunan boşluklar hem de satır atlama ifadesinden dolayı doğru bir şekilde gerçekleşemiyordu. Bu yüzden yorum satırı “fgets” fonksiyonu kullanılarak ilk önce bir diziye atandı daha sonrasında güvenli bir metod olan string kopyalamayı “strncpy” fonksiyonuyla gerçekleştirerek yorum bilgisi depolandı.

Makefile Hatası

Proje dosyalarının tamamının tek bir komutla derlenmesini sağlayan “makefile” dosyası ilk çalıştırıldığında istenildiği gibi sonuçlar vermiyordu. Projede derlenmesi gereken 2 ayrı “main” dosyası olmasına karşın sadece 1 tane “main” dosyası derlenmekteydi. Bu sorun projenin yarısını tamamen işlevsiz hale getiriyordu. Çözüm olarak ise “makefile” dosyası içersine “all” argümanı oluşturularak derlenmesi gereken “main” dosyaları gösterildi.

3. SONUÇLAR

3.1 Efekt Ekleme

- Kullanıcı tarafından girilen efekte göre fotoğraflar düzenlenir. Tek komutla fotoğraf, birden fazla efektlenemez; fakat tek komutla birden fazla fotoğraf, istenilen efekt ile düzenlenebilir. Düzenlenmek istenen fotoğraf veya fotoğrafların yeni halleri, değiştirilmiş isimleriyle aynı klasörün içinde oluşturulur.
- Bazı efektler, bazı fotoğraflarda iyi performans vermeyebilir. Bu performans düşüklüğü, fotoğraflardaki renk dağılımlarının istenilen efekte uygun olmamasından kaynaklanır.
- lena.ascii.pgm fotoğrafı üzerinden beş farklı efektin görüntüsü aşağıda verilmiştir.

Orijinal Fotoğraf

Invert / Noise / Binarize / Contrast / Grey

- Argümanlar, komut satırına girildikten sonra program kullanıcıya geri dönüş sağlar. Bu geri dönüşte ekrana, girilen fotoğrafların başlık bilgileri ve piksel sayıları bastırılır. Aynı zamanda fotoğrafları düzenlemek için okunulan piksel (1 byte) sayıları da kontrol amaçlı ekrana yazdırılır. Aşağıdaki örnekte binarize efektiyle birden fazla fotoğrafın düzenlendiği gözlenmektedir.

Orijinal Fotoğraflar:

casablanca.binary.pgm / barbara.binary.pgm / pepper.ascii.pgm

Input:

```
./pgm_efekt binarize effect_images/barbara.binary.pgm effect_images/pepper.ascii.pgm effect_images/casablanca.binary.pgm
```

Output:

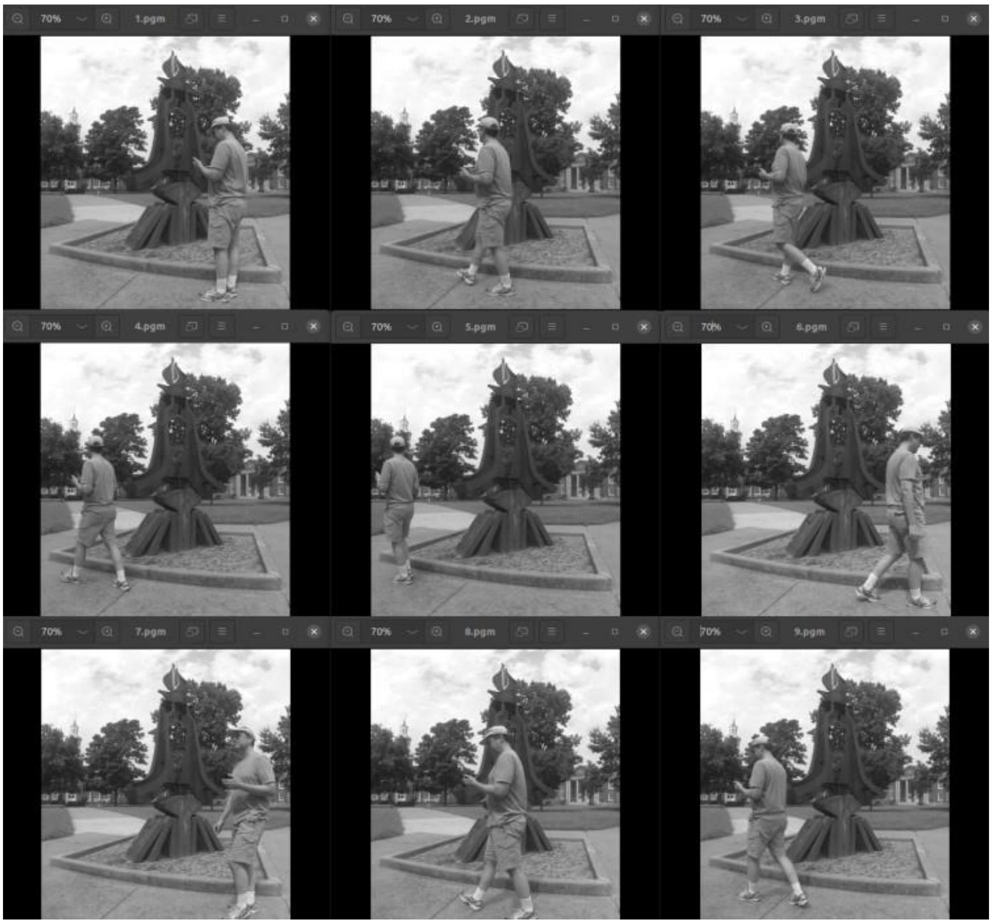
```
Read 262144 bytes. (Should be: 262144)
This is a P5 type PGM image containing 512 x 512 pixels
Read 65536 bytes. (Should be: 65536)
This is a P2 type PGM image containing 256 x 256 pixels
Read 165600 bytes. (Should be: 165600)
This is a P5 type PGM image containing 460 x 360 pixels
```

Sonuç:

casablanca.binary.pgm.binarize / barbara.binary.pgm.binarize / pepper.ascii.pgm.binarize

<u>3.2</u> <u>Medyan Filtreleme</u>

- İskelet kodun içerisine gömülü pgm formatındaki 9 farklı fotoğraf sağ tarafta verilmiştir. Medyan filtrelemedeki amaç, fotoğraflarda bulunan adamı fotoğraftan temizlemektir.
- Kullanıcının ek olarak argüman girmesine gerek yoktur. Bu programı çalıştırmak için exe dosyasının komut satırından çağırılması yeterlidir.
- Program çalıştıktan sonra 9 fotoğraf okunmaya başlar. Bu fotoğrafların başlık bilgileri ve sahip oldukları piksel sayıları ekrana bastırılır. Aynı zamanda program tarafından okunulan piksel (1 byte) sayıları da kontrol amaçlı ekrana yazdırılır.
- Oluşturulan yeni dosya filtered.pgm ismiyle aynı klasörün içine kaydedilir. Kullandığımız 9 fotoğraf üzerinde herhangi bir değişiklik yapılmaz.



1.pgm / 2.pgm / 3.pgm / 4.pgm / 5.pgm / 6.pgm / 7.pgm / 8.pgm / 9.pgm

Input:

```
./pgm_median
```

Output:

```
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
Read 275715 bytes. (Should be: 275715)
This is a P5 type PGM image containing 495 x 557 pixels
```

Sonuç:

filtered.pgm

4. Çıkarımlar ve Gelecek Çalışmalar

- Projedeki temel amaç, pgm formatındaki fotoğrafların düzenlenmesidir. Bu amaç için gerekli aracı, fotoğrafları oluşturan piksellerin tuttuğu değerlerdir. Değerler üzerinde oynama yapılarak fotoğraflara istenilen efektler eklenebilmekte ve filtreleme ile silme işlemi gerçekleşebilmektedir.

- Pgm formatındaki fotoğraflar siyah ve beyazın tonlarından oluştuğu için efektlerin bazı durumlarda düşük performansta çalıştığı gözlemlenebilmektedir. Fakat bu durum, ortalama performans göz önüne alındığında gözardı edilebilir.

- Medyan filtrelemenin çalışabilmesi için birden fazla fotoğraf kullanılmaktadır. Bu fotoğrafların hepsi, aynı başlık bilgilerine ve formata sahip olmalıdır. Aksi takdirde istenilen görüntüyü elde etmek mümkün değildir.

- Binarize efektinin performansı stabil değildir. Bunun sebebi threshold değerinin rastgele atanmasıdır. Performansı stabilize etmek için rastgele atanılan değer gri renk piksel değerine (127) yakın bir şekilde seçilir. Bu durum geliştirilmek istenirse, kullanıcıdan belirli bir orana göre efekt sağlanabilir.

- Efektler sadece pgm formatındaki dosyalar için uygulanabilir. Farklı formattaki fotoğraflar için program çalışmamaktadır. İstenilen formattaki fotoğraf, gerekli işlemler ile pgm formatına dönüştürülebilirse efekt ekleme sağlanabilir.

- Fotoğrafları tek komut ile birden fazla efekt ile düzenlemek mümkün değildir. Bu durum iskelet kodun içerisinde belirli düzenlemeler yapılarak çözülebilir. Fakat piksellerin tuttuğu değerlerin sınırlı bir aralığı olduğu için bozulmaların meydana gelme ihtimali her zaman bulunmaktadır.

- Fotoğraflar komut satırından kolay bir şekilde düzenlenebilmektedir. Fakat bu durum daha da geliştirilerek görsel bir program içerisinden düzenlenebilir.

- Kullanıcının komut satırına yanlış girme ihtimaline karşıt program onu uyarıp kendini sonlandırmaktadır. Geri dönüt olarak doğru kullanım gösterilmektedir.

- Programın efekt çeşitliliği fazla bulunmamaktadır. Bunun sebebi, piksel değerlerindeki değişimlerin nasıl bir performans sergileyeceğinin kesin olarak belli olmamasıdır.