YUV( RAW DATA) ÇÖZÜCÜ PROJESİ

1.Yakup Aytemur-140201066 , 2.Gökhan Ak 140201021

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

1.yakup.aytemur@kuveytturk.com.tr

Özet

*Yazlab1.3 projesinde ki ister olan .yuv uzantılı format videoların açılmasını için gerekli olan çeviri işlemini sağlamaktır. Bu projede çevirme işlemi gerçekleşirken 2 işleme tabii tutulur. 1.olarak .yuv’dan rgb çevirimi 2.olarak ise rgb’den bitmap(bmp) formatına çevirim işlemi gerçekleşir. Çevirme işleminden sonra y-u-v verileri yani renk verilerinden sadece y verisini çekme işlemini bulundurduk. Böylelikle y verisini çektik ve y verisi bize siyah-beyaz görüntü sağlamaktadır. Bu projede kullanılan dil formatı C#’dır.*

# Giriş

Bu projenin amacı .yuv uzatılı dosya verilerini video formatına çevirmektir. .yuv uzatışından çevirilen veriler kaç frame yani kaç kare varsa o kadar fotoğraf karesi oluşturarak sonunda da bu video karelerini birleştirerek basit bir video tasarlamaktır. .yuv’dan çevirip video formatında kayıt yapıp sonuç olarak çok kullanılan video formatlarına çevirmektir.

# Temel Bilgiler

Projede .yuv dan çevirelecek 3 ana format vardır. Bunlar 4:2:2, 4:2:0 , 4:4:4 tür. Bu formatlardan çevirme işlemi farklı farklı olmaktadır. Çünkü hepsi içinde farklı sayıda kare barındırmaktadır. Bizden istenilen sadece y formatına çevirmemizdir. Y formatı siyah-beyazı temsil ederken u ve v değerleri renk kısmını belirtmektedir.

Öncelikli amaç .yuv uzantılı dosyadaki bitleri sırasıyla okumaktır bu okuma işlemi için sadece y kullanılır.

Okuma işleminden sonra rgb çeviri gerçekleşir. Bu işlemden sonra ise bitmape yazma işlemi gerçekleştirmek gerekir.

Başlangıçta belirttiğimiz 3 format içinde farklı okuma işle gerçekleşir çünkü 3 de farklı dosya burundurur.

4:2:0 için 6/4 = 1.5 oran

4:2:2 için 8/4 = 2 oran

4:4:4 için 12/4 = 3 oran bulunmaktadır. Bu oranlara göre işlemler değişmektedirç

# Proje Mimarisi

Proje mimarisinde çalışma mantığı ve projeden kesitler paylaşacağız. Burada işlemler için kısa açıklamalar bulunmaktadır.

## 3.1 Bit Okuma İşlemi

Her format için ayrı ayrı bit okuma işlemi gerçekleştiğinden bir fonksiyon tanımladık ve değişen değeri döndürerek çevirme işleminde çağırırken o değeri kullandık.

Okuma işlemi için :

public void byteReader(double kat)

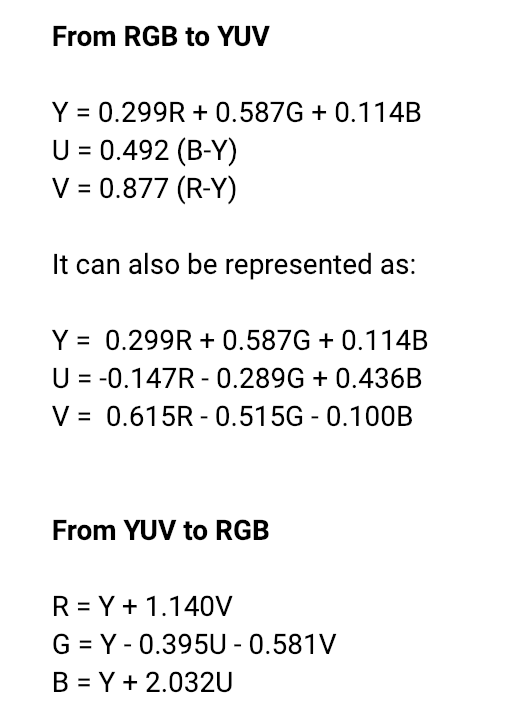
kat değişkeni temel bilgilerde bahsedilen oranları döndürerek Combobox ta seçilen veriye göre oranlar belirlenir ona göre parse işlemi yapılmaktadır.

## 3.2 .yuv’dan Rgb’ye Çevirme

Bura da kullanılacak değerleri internetten matematik işlemlerine bakarak gerçekleştirdik. Değerler genelde sabit olarak bulunmaktadır

static double[,] yuv2rgbmatrisi = new double[3, 3] { { 1, 0, 1.4022 }, { 1, -0.3456, -0.7145 }, { 1, 1.771, 0 } };

yukarıdaki gibi static tanımlayarak aşağıda çağırma işlemleri gerçekleştirdik.



## 3.3 Çevirme Sonucu Bitmap Kayıt

Çevirme işlemi sonucu bitmap olarak dosyaya yazma işlemi gerçekleşir. Burada frame sayısı kadar dosyaya yazma işlemi gerçekleşir.



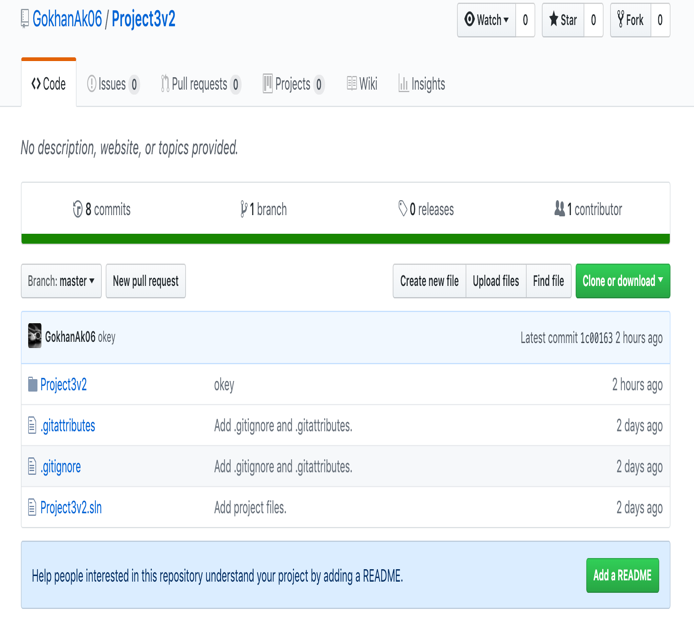
Yukarıdaki görselde ki sadece 1 frame’dir. Bu frameleri birleştirerek biz video elde ediyoruz. Bu video sonra oynatma işlemi gerçekleştiriyoruz.

Kullanıcıdan ayrıyeten .yuv formatı için en ve boy bilgileri istenmektedir. Bu bilgiler alınmak zorundadır çünkü çevirme işleminde çoğunlukla bu bilgiler kullanılmaktadır.

Son olarak ise çevirilen Bitmapler birleştirilerek video oluşması sağlanılmaktadır.

## 3.4 Github Kullanımı

Projede çalışırken iş hayatında proje yönetiminde çokça kullanılan Github uygulamasını kullandık. Bu uygulamayı C#’a indirdik ve ikimizde yazdığımız kodları test etme ve görme imkanı sağladık. Aşağıdaki görselde github bulunmaktadır.



# Sonuçlar

Sonuç olarak bu projede elde ettiğimiz bilgiler ile çevirme işlemindeki matematik işlemlerini fonksiyonları bit okuma bit yazma işlemlerini öğrendik. Burada pek bilgimiz olmadığından başlangıçtan sona kadar internetten araştırma işlemi gerçekleştirdik.

Bu proje ileride daha da çok geliştirilerek yapay zeka alanında görüntü işleme alanında kullanılabilmemize olanak sağladı. İleri C# komutları olan bit okuma ve yazma işlemlerini daha iyi öğrendik.

Algoritma bilgimizi bir üst lige taşıdık denilebilir.

# Kaynakça

1. <https://github.com/GokhanAk06/Project3v2>
2. <https://www.pcmag.com/encyclopedia/term/55166/yuv-rgb-conversion-formulas>
3. <https://stackoverflow.com/questions/44850834/cannot-convert-yuv420-frame-to-bitmap>
4. <https://www.youtube.com/watch?v=4PJbAO-AwwU>
5. <http://www.fourcc.org/fccyvrgb.php>
6. <https://stackoverflow.com/questions/6959275/c-sharp-convert-yuv-422-to-rgb>