|  |
| --- |
| **YILDIZ TEKNİK ÜNİVERSİTESİ – BİLGİSAYAR MÜHENDİSLİĞİ** |
| **Veri Sıkıştırma Projesi** |
| **Görüntü Sıkıştırma** |

|  |
| --- |
| Melike Nur Mermer - 15501010  10.01.2017 |

**Giriş**

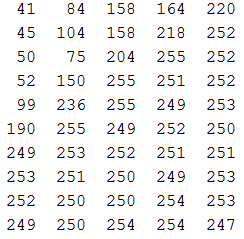
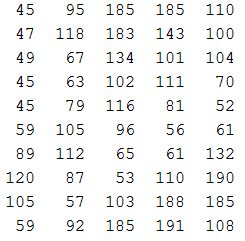
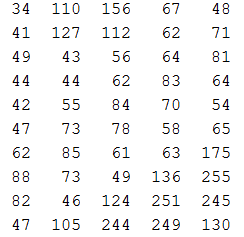
Görüntü sıkıştırma diğer veri sıkıştırma alanlarında olduğu gibi iyi bir seviyede kalite sağlayarak gönderim süresini düşürmeyi amaçlamaktadır. Verinin daha az yer kaplamasını sağlamak için bir miktar bilgi kaybına izin verilmektedir. Bilgi kaybı genellikle renk indirgemesi yapılarak gerçekleşmektedir. Bu çalışmada da K-means yöntemi ile renk indirgemesi yapılarak 24 bitlik görüntüler belirlenen renk sayısına indirgenmiş ardından uygulanan Huffman kodlaması ile entropik kodlama gerçekleştirilmiştir. Sıkıştırılan görüntü simetrik şekilde kodlanarak aynı yöntemle açılması sağlanmıştır. Testte kullanılan 13 görüntü için elde edilen karekök ortalama hatası, peak signal noise ratio (PSNR) değerleri ve sıkıştırma oranları verilmiştir.

**K-means ile renk indirgeme**

Elimizdeki 24 bitlik görüntülerde 224 =16 milyon renk mevcuttur. Görüntünün her bir renk tonu için bir kod atanıp kayıpsız sıkıştırma yapılsa bile sıkıştırılmış verinin de boyutu büyük olacağından sıkıştırma oranı çok küçük olacaktır. Bu sebeple insan gözünün ayırt edemediği birbirine benzeyen bazı renklere aynı kodların atanması ile kayıplı bir sıkıştırma gerçekleştirilmektedir. Burada K-means yöntemi ile renklerin küme merkezleri belirlenmiş ve bu merkezdeki ortalama değer kümedeki bütün renk tonlarını ifade edecek ortak değer olarak atanmıştır. Kümelerin sayısı görüntünün indirgeneceği renk sayısına karşılık gelmektedir. Bu çalışmada indirgenecek renk sayısı 256, 128, 64 ve 32 olarak belirlendiğinden K-means ile oluşturulacak küme sayıları da bu şekilde belirlenmiştir.

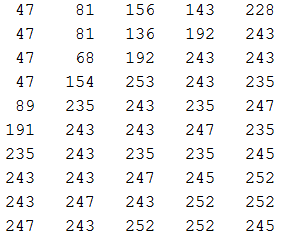
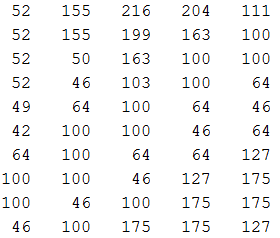
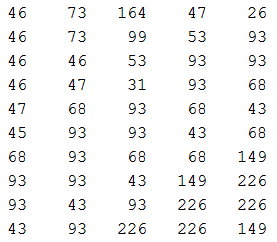
Aşağıda renk indirgeme işlemine örnek olması açısından 64 renge indirgenen “biber” görüntüsünün renk indirgemesi yapılmadan önce ve yapıldıktan sonra aynı bölümüne ait piksel değerleri verilmiştir.

*Renk indirgenmeden önce:*

R G B

*Renk indirgendikten sonra:*

R G B

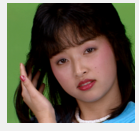
Aşağıdaki şekillerde ise renk indirgemesi yapılmadan önce ve yapıldıktan sonraki görüntülere örnekler verilmiştir:

*biber:*

Orijinal 256 renk 128 renk 64 renk 32 renk

*masuda2:*

Orijinal 256 renk 128 renk 64 renk 32 renk

*ev:*

Orijinal 256 renk 128 renk 64 renk 32 renk

**Huffman kodlama**

Görüntünün indirgendiği renklerin karşılaşılma sıklıkları kullanılarak Huffman ağacı oluşturulmuş ve bu ağaç kullanılarak değişken uzunluklu kodlama işlemi gerçekleştirilmiştir. Belirlenen her renge bir numara verilerek başlanmıştır. Ardından belirlenen renk sayısı kadar elemanı olan frekans dizisi oluşturulmuş (her numara bir renge karşılık geliyor) ve görüntüde bulunan her bir renkteki toplam piksel sayıları bu dizide tutulmuştur. Bu dizinin her elemanı elemanların toplamına bölündüğünde renklerin karşılaşılma olasılıkları çıkarılmaktadır. Bu olasılık dizisi ve karşılık geldiği renkler kullanılarak Huffman ağacı oluşturulur. Huffman ağacını oluşturan algoritma ağaçla beraber her renge karşılık gelen kodların tutulduğu bir tablo döndürmektedir. Örneğin 32 renge indirgenmiş “biber” görüntüsünün her renginin piksel frekansları ve kodlanacağı bit stringleri aşağıda verilmektedir.

*32 renge indirgenmiş görüntünün her renginin piksel frekansları:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| 1017 | 808 | 396 | 822 | 553 | 310 | 2708 | 423 | 765 | 130 | 901 | 37 | 1348 | 134 | 525 | 439 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** |
| 291 | 138 | 57 | 817 | 245 | 274 | 62 | 44 | 1086 | 231 | 296 | 332 | 330 | 110 | 536 | 219 |

Bu tablodan anlaşıldığı gibi en yüksek frekanslar 7, 13, 25, 1 no’lu renklere aittir. En çok karşılaşılan bu renklerin en az bitle ifade edilmesi gerekmektedir.

*Huffman ağacına göre renklerin kodlandığı bit stringleri:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **'9'** | **'3'** | **'12'** | **'24'** | **'30'** | **'32'** | **'2'** | **'20'** |
| '0000' | '00010' | '00011000' | '00011001' | '0001101' | '000111' | '0010' | '0011' |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **'4'** | **'8'** | **'16'** | **'11'** | **'26'** | **'21'** | **'19'** | **'23'** | **'10'** | **'14'** |
| '0100' | '01010' | '01011' | '0110' | '011100' | '011101' | '01111000' | '01111001' | '0111101' | '0111110' |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **'10'** | **'14'** | **'18'** | **'1'** | **'15'** | **'31'** | **'25'** | **'5'** |
| '0111101' | '0111110' | '0111111' | '1000' | '10010' | '10011' | '1010' | '10110' |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **'22'** | **'17'** | **'27'** | **'6'** | **'29'** | **'28'** | **'13'** | **'7'** |
| '101110' | '101111' | '110000' | '110001' | '110010' | '110011' | '1101' | '111' |

Buradan anlaşıldığı gibi frekansı yüksek olan renkler daha kısa kodlarla, frekansı düşük olan renkler daha uzun kodlarla ifade edilmektedir.

Sonuçta görüntünün piksellerinin her biri karşılık gelen kod değeri ile kodlandığında bir “code” bit stringi elde edilmektedir. Bu stringin uzunluğu sıkıştırılmış görüntünün boyutunu vermektedir. Sıkıştırma oranı hesaplanırken çıkış büyüklüğü olarak bu değer kullanılmıştır.

Sıkıştırılmış görüntünün geriye açılması işlemi sırasında aynı adımlar tersten işlenir. Eldeki “code” dizisinden her pikselin renk karşılığı elde edilir. Bu renkler bir matrise yazılarak görüntüye çevrilir ve açılmış (decompressed) görüntü elde edilir.

**Sıkıştırma başarısının hesaplanması**

Bu çalışmada kayıplı sıkıştırmada başarı ölçümü için karekök ortalama ve Peak Signal Noise Ratio (PSNR) kullanılmıştır. Sıkıştırmanın boyutunu ölçmek için ise sıkıştırma oranı (compression ratio) kullanılmıştır. Sıkıştırma oranı hesaplanırken çıkış verisi olarak Huffman ile kodlanan bit stringinin uzunluğu, giriş verisi olarak ise 24 bitlik görüntü kullanıldığı için görüntünün toplam piksel miktarı\*24 alınmıştır. Kullanılan ölçütlerin denklemleri aşağıda verilmektedir.

erms =

PSNR = 10log10

Compression Ratio =

Bu ölçütler kullanılarak elimizdeki 13 adet 24 bitlik görüntüye önce K-means ile kayıplı sıkıştırma, sonra Huffman ile kayıpsız değişken uzunluklu kodlama uygulanmıştır. Uygulama sonucunda elde edilen erms değerlerinin gösterildiği tablo ve grafik aşağıdadır.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **erms** | **256** | **128** | **64** | **32** |
| **biber** | 3,37 | 4,58 | 6,33 | 9,23 |
| **boat** | 2,77 | 3,61 | 4,65 | 6,16 |
| **boats** | 2,80 | 3,12 | 4,34 | 6,12 |
| **ev** | 2,09 | 2,38 | 3,01 | 4,18 |
| **goldhill** | 2,46 | 3,21 | 4,49 | 6,34 |
| **masuda1** | 1,96 | 2,71 | 3,74 | 5,14 |
| **masuda2** | 2,04 | 2,76 | 3,65 | 5,00 |
| **plast** | 4,60 | 5,99 | 7,92 | 10,04 |
| **tiffany** | 1,93 | 2,22 | 2,96 | 3,92 |
| **ucak** | 2,08 | 2,18 | 3,14 | 4,20 |
| **zelda** | 3,01 | 3,77 | 5,43 | 8,11 |
| **barbara** | 3,53 | 4,29 | 5,83 | 8,14 |
| **magnify** | \* | 207,46 | 207,46 | 207,46 |

Burada görüldüğü gibi 256 renge indirgenen görüntülerin erms değerleri daha düşük olmaktadır. Yani 256 renkle ifade edilebilen görüntülerin gerçek görüntüye benzerliği daha yüksektir.

Görüntü kalitesini belirleyen en önemli ölçüt PSNR’dir. Aşağıda farklı renk sayılarına indirgenmiş görüntülerin PSNR değerlerini gösteren bir tablo ve grafik mevcuttur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **psnr** | **256** | **128** | **64** | **32** |
| **biber** | 37,6 | 28,8 | 20 | 10,5 |
| **boat** | 39,3 | 30,9 | 22,6 | 14 |
| **boats** | 39,2 | 32,2 | 23,2 | 14,1 |
| **ev** | 41,7 | 34,5 | 26,4 | 17,4 |
| **goldhill** | 40,3 | 31,9 | 22,9 | 13,8 |
| **masuda1** | 42,3 | 33,4 | 24,5 | 15,6 |
| **masuda2** | 41,9 | 33,3 | 24,7 | 15,8 |
| **plast** | 32,7 | 26,5 | 18 | 9,79 |
| **tiffany** | 42,4 | 35,1 | 26,5 | 17,9 |
| **ucak** | 41,8 | 35,3 | 26 | 17,4 |
| **zelda** | 38,5 | 30,5 | 21,3 | 11,6 |
| **barbara** | 37,2 | 29,4 | 20,7 | 11,6 |
| **magnify** | \* | -4,26 | -10,4 | -16,5 |

Bu grafikten anlaşıldığı gibi 256 renge indirgenen görüntülerin kalitesi oldukça yüksek olmaktadır. İndirgenecek renk sayısı azaldıkça görüntü kalitesi de düşmektedir. PSNR 30-35 seviyelerinde olması kabul edilebilir bir kayıp olarak düşünüldüğünde görüntülerin birçoğunu 128 renge indirgeyerek de başarılı bir sıkıştırma elde edilebilir. Ancak renk tonlarının çeşitliliğinin fazla olduğu biber, plast, barbara gibi görüntülerde 128 renge indirgeme doğru bir seçim olmayacaktır.

İndirgenen renk sayılarına göre görüntülerin PSNR değerlerini gösteren bir grafik aşağıda sunulmuştur.

Bu grafiğe bakıldığında renk çeşitliliği az olan uçak, ev, tiffany görüntüleri 128, 64 ve 32 renge indirgendiğinde diğer görüntülere göre daha yüksek PSNR’ler elde edilmiştir. Yine benzer mantıkla renk çeşitliliği en çok olan plast görüntüsü için 256 renge indirgemek bile ancak 32,7 gibi bir PSNR sağlamaktadır.

Huffman ile kodlanan görüntünün bit stringinin görüntünün boyutuna oranını ifade eden sıkıştırma başarısı değerleri ise aşağıdaki tablo ve grafikte verilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **c.ratio** | **256** | **128** | **64** | **32** |
| **biber** | 0,29 | 0,26 | 0,22 | 0,18 |
| **boat** | 0,32 | 0,28 | 0,24 | 0,20 |
| **boats** | 0,31 | 0,27 | 0,23 | 0,19 |
| **ev** | 0,29 | 0,25 | 0,21 | 0,17 |
| **goldhill** | 0,32 | 0,28 | 0,24 | 0,20 |
| **masuda1** | 0,30 | 0,26 | 0,23 | 0,19 |
| **masuda2** | 0,28 | 0,24 | 0,21 | 0,16 |
| **plast** | 0,31 | 0,27 | 0,23 | 0,19 |
| **tiffany** | 0,32 | 0,28 | 0,23 | 0,19 |
| **ucak** | 0,29 | 0,26 | 0,21 | 0,18 |
| **zelda** | 0,32 | 0,28 | 0,24 | 0,20 |
| **barbara** | 0,31 | 0,28 | 0,23 | 0,20 |
| **magnify** | \* | 0,35 | 0,31 | 0,16 |

Sıkıştırma oranının yüksek olması demek çıkış stringinin büyük olması demek olduğundan bu ölçütün küçük olması istenmektedir. Bu grafikte de görüldüğü gibi 256 renge indirgenen görüntülerde daha çok renk çeşitliliği olduğundan çıkışta kodlanan bit stringleri de uzun olmaktadır. Bu sebeple 256 renkli görüntülerin sıkıştırma oranı daha az renkli görüntülerin sıkıştırma oranından yüksek çıkmaktadır. Burada 256 renge indirgendiğinde 0,29 gibi bir sıkıştırma oranı elde edilen biber, ev, uçak görüntülerine bakıldığında bu görüntülerden renk çeşitliliği az olanların daha az renge indirgendiğinde daha düşük sıkıştırma oranlarına ulaştığı görülmektedir. (ev 128, 64 ve 32 renge indirgendiğinde diğerlerinden daha düşük, uçak 64 renge indirgendiğinde diğerlerinden daha düşük.)

**Sonuç**

Bu çalışmada kayıplı (K-means) ve kayıpsız (Huffman) sıkıştırma işlemleri görüntü verileri üzerinde uygulanmış ve elde edilen sıkıştırma belirli ölçütlere göre analiz edilmiştir. Test görüntülerinde alınan sonuçlara göre görüntünün daha çok sayıda renge indirgenmesi sıkıştırılan görüntünün kalitesini artırmaktayken sıkıştırma oranını da artırmaktadır. Yani sıkıştırma işlemi sonunda daha yüksek boyutlu çıkışlar oluşmaktadır. Görüntünün az sayıda renge indirgenmesi ise sıkıştırma oranını düşürürken görüntü kalitesini de düşürmektedir. Görüntü sıkıştırma işleminde bu iki kriter esas alınarak indirgenecek renk sayısına karar verilmelidir. Test verisi olarak kullanılan 24 bitlik görüntülerin birçoğunda 128 renk iyi bir seviyede görüntü kalitesi (PSNR) sağlarken sıkıştırma oranları da iyi seviyelerdedir. Ancak renk çeşitliliği fazla olan biber, plast, barbara gibi görüntülerde 128 renge indirgemek kalite açısından yeterli olmayabilir.