

ANAHTAR KARE ÇIKARIMI

YZ 5509, ZEKİ OPTİMİZASYON YÖNTEMLERİ

Okan İhsan Bağrıaçık
okanihsanbagriacik@posta.mu.edu.tr

Friday 14th January, 2022

Abstract

Anahtar kareler, video açıklamalarında önemli bir rol oynar. Video soyutlama için yaygın olarak kullanılan yöntemlerden biridir, çünkü bu, yeterli içerik gösterimi ile geniş bir video verisi setini daha hızlı bir şekilde işlememize yardımcı olacaktır. Projede, entropi değeri kullanılarak anahtar kare çıkarımı ve aynı zamanda öklid değerlerine göre kare çıkarımı adına 2 model yapılmıştır. Önerilen yaklaşım, çerçeveleri genel özellik olarak entropi değerlerine göre sınıflandırır ve temsili anahtar çerçeve olarak her sınıftan çerçeve seçer. Ayrıca, yerel özellik olarak entropi değerini kullanarak seçilen ana karelerdeki gereksiz kareleri ortadan kaldırır. Öklid uzaklık mesafesine göre aynı işlem yapılmaktadır. Sonuçlar bir dosyada atılarak GIF oluşturulmaktadır ve kareler tek tek görülmektedir.

Anahtar Kelimeler : Anahtar Kare Çıkarması, Öklid Uzaklık, Entropi Değeri, GIF

1 Giriş

Son yıllarda, video içerik yönetimine yönelik yazılım araçlarındaki gelişmeler, video içeriğinin verimli bir şekilde sınıflandırılmasını mümkün kıldı ve video arama ve alma sürecini daha da hızlı hale getirdi. Mevcut araştırma topluluğu, insan etkileşimi, yani zaman tüketimi ve insan bakış açısı içeren sistemin dezavantajlarının üstesinden gelmek için video içerik yönetiminin otomasyonuna odaklanmaktadır.

Video segmentasyonu ve ana kare çıkarma, video analizinin ve içerik tabanlı video alımının temelidir. Anahtar kare çıkarma, video analizi ve yönetiminin önemli bir parçasıdır ve video indeksleme, tarama ve geri alma için uygun bir video özeti sağlar. Anahtar karelerin kullanımı, video indekslemede gereken veri miktarını azaltır ve video içeriğiyle ilgilenmek için çerçeve sağlar.

Video, verilerin görsel temsili olarak tanımlanabilir. Ham video sahneler dizisi, sahneler çekimler dizisi ve çekimler kareler dizisi olarak yapılandırılabilir. Sunulan çalışmaların çoğu, segmentasyon ve ana kare çıkarma için bu video yapısından yararlanır. Anahtar kare, çekimin göze çarpan içeriğini ve bilgilerini temsil edebilen karedir. Çıkarılan anahtar kareler, videonun özelliklerini özetlemelidir ve bir videonun görüntü özellikleri, zaman dizisindeki tüm anahtar kareler tarafından izlenebilir. Anahtar çerçevelerin seçimi için birçok yöntem geliştirilmiştir. Bir geri alma uygulamasında, bir video dizisi, zaman içinde, her biri benzer içerik içeren bir dizi daha kısa parçaya bölünür. Bu segmentler aranan veri miktarını büyük ölçüde azaltan temsili anahtar karelerle temsil edilir. Ancak anahtar kareler, segment içindeki nesnelerin hareketlerini ve eylemlerini tanımlamaz. Sahnelerin ana karelerini seçmek, aynı zamanda fazlalık olabilecek diğer kareleri hariç tutarken, içerik varyasyonlarının çoğunu yakalamamızı sağlar. Sahnedeki geri kalan tüm kareler, ilk karenin mantıksal ve sürekli uzantıları olarak kabul edilebileceğinden,

ilk kareyi seçmek doğal bir seçim gibi görünüyor, ancak sahnedeki tüm kareler için en iyi eşleşme olmayabilir. .

Anahtar kare çıkarımı, video bilgi erişiminde önemli araştırma konularından biri olarak kabul edilmiştir. Ana çerçeve çıkarmada ilerleme kaydedilmiş olmasına rağmen, mevcut yaklaşımlar ya hesaplama açısından pahalıdır ya da göze çarpan görsel içeriği yakalamada etkisizdir. Pojede 3 farklı kod sayfasından oluşmaktadır. Bir kod sayfası video kareleri oluşturmak için kullanılmıştır. Lokal kameradan video çeken kod 300 kare yakalamaktadır. Daha sonrasında Entropi ve Öklidyen değerlerine göre kareler Diferansiyel algoritma ile işleminden geçmektedir. Algoritma sonucunda en ideal kareler seçilmektedir ve hem kareler tek tek incelenebilir hale gelmektedir hemde GIF'leri oluşturulmaktadır.

2 İçerik

Projemizde evrimsel algoritmalarından Diferansiyel Evrim algoritması kullanılmıştır. Bu algoritma içerisinde karelerden ideal olanları seçilerek mutasyon, çaprazlama ve seleksiyon işlemleri yapılmaktadır. Bu işlemler sonucunda video içerisinde ideal anahtar kareler çıkarılmaktadır.

2.1 Anahtar Kare ve Video İnceleme

Son yıllarda, video içerik yönetimine yönelik yazılım araçlarındaki gelişmeler, video içeriğinin verimli bir şekilde sınıflandırılmasını mümkün kılmış ve video arama ve alma sürecini daha da hızlı hale getirmiştir. Mevcut araştırma topluluğu, insan etkileşimi, yani zaman tüketimi ve insan perspektifi içeren sistemin dezavantajlarının üstesinden gelerek, video içerik yönetiminin otomasyonuna odaklanmaktadır.

Video segmentasyonu ve ana kare çıkarma, video analizinin ve içerik tabanlı video alımının temelidir. Anahtar kare çıkarma, video analizi ve yönetiminin önemli bir parçasıdır ve video indeksleme, tarama ve geri alma için uygun bir video özeti sağlar. Anahtar karelerin kullanımı, video indekslemede gereken veri miktarını azaltır ve video içeriğiyle ilgilenmek için çerçeve sağlar. Video, verilerin görsel temsili olarak tanımlanabilir. Ham video sahneler dizisi, sahneler çekimler dizisi ve çekimler kareler dizisi olarak yapılandırılabilir. Sunulan çalışmaların çoğu, segmentasyon ve ana kare çıkarma için bu video yapısından yararlanır. Anahtar kare, çekimin göze çarpan içeriğini ve bilgilerini temsil edebilen karedir. Çıkarılan anahtar kareler, videonun özelliklerini özetlemelidir ve bir videonun görüntü özellikleri, zaman dizisindeki tüm anahtar kareler tarafından izlenebilir. Anahtar çerçevelerin seçimi için birçok yöntem geliştirilmiştir. Bir geri alma uygulamasında, bir video dizisi, zaman içinde, her biri benzer içerik içeren bir dizi daha kısa parçaya bölünür. Bu segmentler aranan veri miktarını büyük ölçüde azaltan temsili anahtar karelerle temsil edilir. Ancak anahtar kareler, segment içindeki nesnelerin hareketlerini ve eylemlerini tanımlamaz. Sahnelerin ana karelerini seçmek, aynı zamanda fazlalık olabilecek diğer kareleri hariç tutarken, içerik varyasyonlarının çoğunu yakalamamızı sağlar. Sahnedeki geri kalan tüm kareler, ilk karenin mantıksal ve sürekli uzantıları olarak kabul edilebileceğinden, ilk kareyi seçmek doğal bir seçim gibi görünüyor, ancak sahnedeki tüm kareler için en iyi eşleşme olmayabilir.

2.2 Video Segmentasyonu

Video, keskin geçiş kesme algılaması kullanılarak çekim sınırlarının saptanmasına dayalı olarak çekimlere bölünür. Kesim, bir atış ile bir sonraki atış arasındaki keskin geçiş olarak tanımlanır. Kesimler genellikle iki ardışık görüntü için renk ve parlaklık düzeninde ani bir değişikliğe karşılık

gelir. Bu yaklaşımın arkasındaki ilke, bir çekimdeki iki ardışık karenin arka planında ve nesne içeriğinde önemli ölçüde değişmemesi nedeniyle, genel renk ve parlaklık dağılımlarının çok az farklı olmasıdır. Öte yandan, arka planın renginde ve aydınlatmasında çarpıcı bir değişikliğin olduğu bir sahnemiz varsa, görüntünün renk düzeyi üzerinde sonuçlar doğuracaktır, bu da çekimin nesnelerinde ve arka planında değişiklik olduğunu gösterir. Kesik tespiti ile videonun çekimlere bölünmesi Histogram farklılıkları, Şablon eşleştirme, Kenar değişim oranı vb. literatürde önerildiği gibi birçok yöntemle yapılabilir. Önerilen çalışma, videoyu segmentlere ayırmak için şablon eşleştirme algoritmasını kullanır. Bu yöntemde, videonun ardışık iki karesi piksel piksel karşılaştırılır ve iki kare arasındaki bir ko-ilişki faktörü hesaplanır. Ko-ilişki faktörü eşikten, yani 0,9'dan küçükse, videoda bir kesim vardır ve video çekimlere bölünür. Şablon eşleştirme ile bölütlemenin, eklemenin neden olduğu anlamsız kısa çekimlerle sonuçlandığı gözlemlenmiştir. fotoğrafik etki (örneğin, kaybolur ve çözülür). Bu solma ve çözülme çekimleri, çekimin minimum zaman çerçevesi sabitlenerek diğer çekimlerle birleştirilebilir.

2.3 Entropi Değeri

Gri seviye sayısının 256 olarak nicelendiği bir video dizisinden tipik bir kare düşünün. $h_f(k)$ kare f 'nin histogramı ve k gri seviye olsun, öyle ki $0 < k < 2^b - 1$, burada b is, görüntü nicelendirme seviyelerinin temsil edilebildiği bit sayısı. Video karesi M sınıfı satır N sütun ise, bu gri seviyenin çerçevede görünme olasılığı;

$$p_f(k) = h_f(k) / (M.N)$$

Çerçevenin entropisi, ters görünüm olasılığı $p_f(k)$ ile görünüm olasılığı $p_f(k)$ arasındaki logun çarpımının toplamı olarak tanımlanabilir.

$$Entropy = - \sum_{k=0}^{k=\max} \log(p_f(k)) / p_f(k)$$

Entropi kutuları arasındaki mesafeyi artırmak, böylece değişen entropi değerlerine sahip her bir çerçevenin sınıflandırılması kolaylıkla yapılabilir, kare entropi değerlerinin ayrık değeri dikkate alınır (değiştirilmiş entropi değeri).

$$En_{mf} = round(En_f^2)$$

En_{mf} entropi değeri En_f ile çerçeve f 'nin entropi değeri değiştirilir. Algoritma 1 anahtar kare çıkarımı, her kare için değiştirilmiş entropi değerini hesaplar ve bunları sınıflandırır. Yeni kutular, yeni entropi değerleri hesaplanırken başlatılır. Burada En_{mf} , çerçeve f 'nin entropi değeri En_f ile değiştirilir. Algoritma 1 anahtar kare çıkarımı, her kare için değiştirilmiş entropi değerini hesaplar ve bunları sınıflandırır. Yeni depolar, yeni entropi değerleri hesaplandığında başlatılır.

2.4 Benzer Karelerin Çıkarılması

Video klabin farklı çekimlerinde birçok kez nesne ve arka planın tekrarlandığı gözlemlenmiştir, örn. haber hikayesi anlatan bir haber okuyucusu. Bu, bir veya daha fazla gereksiz anahtar kareye yol açar. Bu fazlalık anahtar kareleri ortadan kaldırmak için, bir filtreleme adımı gerçekleştirilir, burada her bir anahtar kare, kopya veya kopyaya yakın kareleri bulmak için diğer tüm ana karelerle karşılaştırılır. İki çerçeve arasındaki benzerliği bulmak için parçalı entropi tekniği uygulanır.

Parçalı entropi tekniğinde her çerçeve 64 parçaya bölünür ve her parçanın entropisi ayrı ayrı hesaplanır. Bu teknikte entropi yerel özellik olarak kullanılır. Bu yöntem sayesinde, iki çerçeve arasındaki varyasyon, tüm çerçevenin entropisi ile karşılaştırıldığında daha doğru tanımlama sağlayan segment düzeyinde izlenebilir. İki çerçevenin farklılığını ölçmek için, iki çerçevenin parçalı entropilerinin farkının standart sapması hesaplanır. Standart sapma Sıfır değerine yaklaşıyorsa, iki çerçeve benzer olarak kabul edilir, ikinci kare, yinelenen kare olarak ortadan kaldırıldı.

2.5 Öklidyen Değeri

Tekrar eden kareleri bulmak amacıyla, video olarak sınıflandırılmış her anahtar film karesi için, diğer anahtar film kareleri arasında bunlara benzer olanlar aranmaktadır. Bu benzerlik için resimlerin RGB değerleri üzerinden Öklid uzaklığı baz alınmaktadır. Bu değer üzerinden bir eşik değeri kullanılarak en yakın tekrarlar bulunmektedir. Bazı projelerde makine öğrenmesi algoritması olan KNN kullanılarak buradan alınan öklid değerlerini ile yakın değerlerde bulunabilmektedir.

2.6 Diferansiyel Evrim Algoritması

Farksal Gelişim Algoritması Price ve Storn tarafından 1995 yılında geliştirilmiş, özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, isleyiş ve operatörleri itibariyle genetik algoritmaya dayanan popülasyon temelli sezgisel optimizasyon tekniğidir.

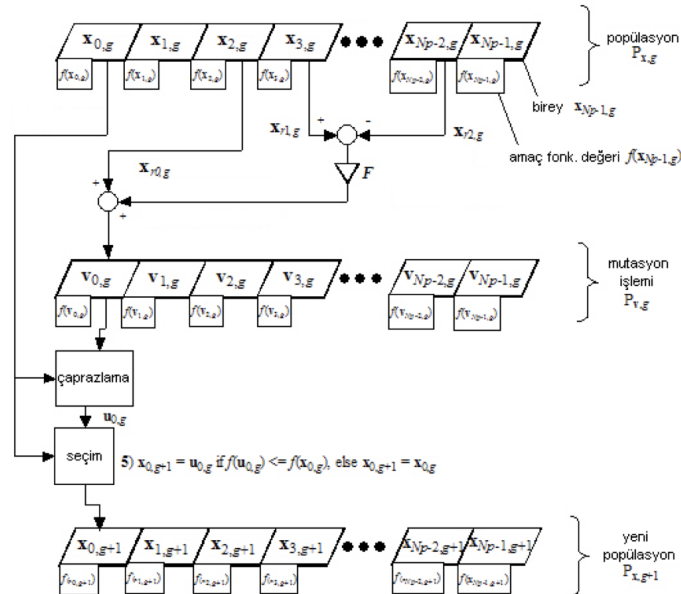


Figure 1: Diferansiyel Evrim Algoritması İşleyişi

Temel olarak, DE algoritması popülasyon içerisinde rastgele seçilen iki bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır. Literatürde DE algoritmasının bu işleminin farklı çeşitleri bulunmaktadır: • en iyi birey / 1 fark vektörü / üstel çaprazlama

- rastgele bireyler / 1 fark vektörü / üstel çaprazlama
- rastgele ve en iyi bireyler / 1 fark vektörü / üstel çaprazlama
- en iyi birey / 2 fark vektörü / üstel çaprazlama
- rastgele bireyler / 2 fark vektörü / üstel çaprazlama
- en iyi birey / 1 fark vektörü / binom çaprazlama
- rastgele bireyler / 1 fark vektörü / binom çaprazlama
- rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlama
- en iyi birey / 2 fark vektörü / binom çaprazlama
- rastgele bireyler / 2 fark vektörü / binom çaprazlama

Bu genel gösterim tarzında, ilk kısım popülasyon içinden seçilen bireyleri, ikinci kısım seçilen bireylerin oluşturacağı fark vektörünün sayısını, son kısım ise çaprazlama tipini belirtmektedir. Bu çalışmada pratikte en sık kullanılan iki stratejisi rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlama ile rastgele bireyler / 1 fark vektörü / binom çaprazlama olarak seçilmiştir. DE algoritması 3 önemli kontrol parametresine sahiptir: skala faktörü (SF), çaprazlama olasılık katsayısı (CR) ve popülasyon boyu (PS). DE algoritmasının başında PS optimizasyon parametrelerine göre belirlenir ve optimizasyon sürecinde değiştirilmez. DE algoritması 3 temel operatöre sahiptir: mutasyon, çaprazlama ve seçim. Mutasyon ve çaprazlama operatörleri yeni bireyler üretir ve seçim operatörü ile uygun olanlar belirlenir, bu şekilde popülasyonda sürekli en iyi bireylerin bulunması sağlanır.

2.6.1 Mutasyon

Her iki DE algoritma stratejisi için mutasyona uğramış bireyler aşağıdaki şekilde bulunur:

$$v_{i,g+1} = x_{i,g} + SF(x_{b,g} - x_{i,g}) + SF(x_{r1,g} - x_{r2,g})$$

$$v_{i,g+1} = x_{r3,g} + SF(x_{r1,g} - x_{r2,g})$$

Bu eşitliklerde $r1, r2, r3 \in [1, PS]$ rastgele katsayılarıdır. popülasyon içerisindeki en iyi maliyet fonksiyonuna sahip bireyi göstermektedir. Rastgele katsayılar birbirinden farklı olmak zorundadır. Skala faktörü (SF) sabit bir değerdir ve $0 < SF < 2$ olarak değişir. Skala faktörü rastgele seçilen bireylerin fark vektörlerinin yükseltilmesinin kontrolünde kullanılır.

2.6.2 Çaprazlama

Yeni bireyler çaprazlama operatörü ile aşağıdaki şekilde bulunabilir:

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{eğer } r \leq CR \\ x_{i,g}, & \text{eğer } r > CR \end{cases}$$

Bu denklemde $r \in [0, 1]$ rastgele sayı üreticini ve deneme bireylerini göstermektedir. Çaprazlama sabiti (CR) $E \in [0, 1]$ genellikle kullanıcı tarafından belirlenir. Eğer $CR = 1$ ise, oluşacak her yeni birey mutant olarak belirlenir, yani rastgele seçilen fark vektörüne () göre seçilir. Diğer taraftan $CR = 0$ durumunda, tüm bireyler önceki jenerasyondan (gelir. Diğer bir deyişle, popülasyonda değişim olmaz, yeni bireyler oluşmaz. Rastgele sayı üretici, mutant bireylerden en az birinden bir sonraki jenerasyona birey üretilmesini garanti eder.

2.6.3 Seleksiyon

Bir sonraki jenerasyondaki bireylerin tespitinde seçim operatörünün işletilmesi sırasında, deneme bireyinin maliyet fonksiyonu değeri hedef bireyin maliyet fonksiyonu ile karşılaştırılır ve sonraki jenerasyona optimizasyon probleminde göre yeni birey seçilir. Aşağıdaki eşitlikte minimum problemi için seçim operatörünün denklemi verilmiştir:

$$x_{i,g+1} = \begin{cases} u_{i,g+1}, & \text{eğer } f(u_{i,g+1}) < f(x_{i,g}) \\ x_{i,g}, & \text{diğer durumlar} \end{cases}$$

Burada f maliyet fonksiyonunu göstermektedir. Eğer deneme bireyi hedef bireyden daha iyi bir maliyet fonksiyonuna sahipse, bir sonraki jenerasyonda kullanılacak birey deneme bireyi olarak seçilir. Aksi durumda popülasyondaki hedef bireyde değişiklik olmaz.

3 Literatür Araştırması

Anahtar çerçeve çıkarmanın temel bir kuralı, anahtar çerçeve çıkarmanın yetersiz olmaktan çok yanlış olmasıdır. Bu nedenle, çıkarma sırasında tekrarlayan veya fazla bilgi içeren çerçevelerin atılması gerekir [1]. Mevcut segmentasyon ve anahtar kare çıkarma algoritmaları, çekim tabanlı segmentasyon ve nesne tabanlı segmentasyon olarak da bilinen zamansal tabanlı segmentasyon olarak sınıflandırılabilir.

3.1 Çekim Tabanlı Video Segmentasyonu

Çekime dayalı video bölümlenme, genellikle iki ana adımın, yani zamansal bölümlenmenin ve ana kare çıkarmanın dahil olduğu bir veri soyutlama süreci olarak düşünülebilir. Zamansal bölümlenme, renk düzeni [2], entropi [3], [4] vb. gibi bir veya birkaç karesel özelliği kullanarak bir video dizisini bir dizi video çekimine sınıflandırır. anahtar kareler kümesi. Genellikle, bir video görüntüsünü birkaç kümeye bölen ve ana kareler olarak kümelerin merkezini seçen tipik bir kümeleme işlemi olarak modellenir. Her çekimin anahtar kareleri, K-ortalamlar yöntemi [2] kullanılarak çıkarılır. [5]'te Gauss karışım modeli (GMM), RGB renk uzayındaki renk histogramlarının zamansal varyasyonunu modellemek için kullanılır. Çekimde bulunan çerçeveler, özellik temsiline göre birkaç kümeye ayrılabilir. Her küme için, merkeze en yakın kare anahtar kare olarak seçilir. Küme sayısı Bayesian Information Criterion ile belirlenebilir. Bu yöntemin ana dezavantajı, kümelerin sayısını otomatik olarak belirleyememesi ve dolayısıyla kümelemeyi video içeriğine otomatik olarak uyarlayamamasıdır.

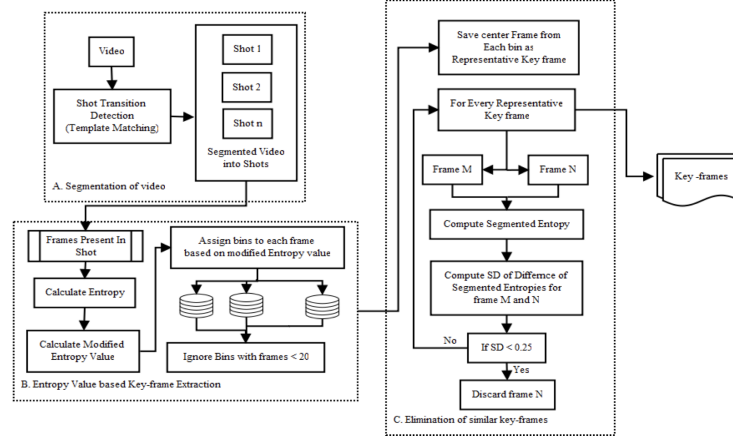


Figure 2: Sistem İşleyiş Diagramı

3.2 Nesne tabanlı video segmentasyonu

Nesne tabanlı video segmentasyonu, bir video görüntüsünü, genellikle uygulamaya bağlı olan nesnelere ve arka plana ayırmaktır. Temel birim olarak bir çerçeveye sahip olan çekim tabanlı video segmentasyonundan farklı olarak, nesne tabanlı segmentasyon, ham videoyu daha yüksek anlamsal düzeyde temsil eden nesneler sağlayabilir. Nesne tabanlı video segmentasyonu, bir video dizisini, her biri bir videodaki zamansal ve/veya uzamsal özelliklerle temsil edilen bir model olarak kabul edilebilecek birkaç nesneye sınıflandırır. Mevcut nesne tabanlı video bölütleme yöntemleri üç tipte sınıflandırılabilir: uzaysal öncelikli bölütleme, zamansal öncelikli bölütleme ve birleşik uzaysal ve zamansal bölütleme. Daha yeni ilgi alanları, uzaysal ve zamansal alanlarda ortaklaşa göze çarpan video yapılarını tanıyan insan görüşünün doğasından dolayı ortak uzaysal ve zamansal video segmentasyonu üzerinedir [6]. Bu nedenle, nesne segmentasyonu için çok boyutlu bir özellik alanı oluşturmak için hem uzaysal hem de zamansal piksel bazında öznetelikler çıkarılır. Renk histogramı gibi kare tabanlı özellikleri kullanan ana kare çıkarma yöntemleriyle karşılaştırıldığında, bu yaklaşımlar genellikle hesaplama açısından daha pahalıdır.

4 Uygulama

Proje genel olarak 3 kod sayfasından oluşmaktadır. Elimizde veri olarak video kareleri bulunmadığından dolayı önce Python dilinde video karesi oluşturmak için bir kod yazılmıştır. Bu kod sayfasında lokal kamera açılmaktadır ve bir video çekmektedir. Daha sonrasında bu videoyu 300 video karesine bölmektedir. Bunu yaptıktan sonra yine lokalde bir dosya içerisine bütün kareleri veri olarak atmaktadır.

```

import cv2

|
WRITE_BUFFER_LOCATION = "/Users/okanbagriacik/Desktop/Kareler/"

NUMBER_OF_FRAMES = 300

cap = cv2.VideoCapture(0)
# Capture first frame.
ret0, frame0 = cap.read()
init_gray = cv2.cvtColor(frame0, cv2.COLOR_BGR2GRAY)
cv2.imwrite(WRITE_BUFFER_LOCATION + str(0) + ".jpg", init_gray)
i = 1

for j in range(0, NUMBER_OF_FRAMES):
    # Capture second frame.
    ret, frame1 = cap.read()

    # Convert frames to grayscale. (ie. previous and current frames)
    gray0 = cv2.cvtColor(frame0, cv2.COLOR_BGR2GRAY)
    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)

    # Display the current frame.
    cv2.imshow('frame1',gray1)
    cv2.imwrite(WRITE_BUFFER_LOCATION + str(i) + ".jpg", gray1)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    # Set previous frame.
    ret0, frame0 = cap.read()
    i += 1

# When everything done, release the capture

cap.release()
cv2.destroyAllWindows()

```

Figure 3: Video Kare Oluşturma Ptyhon Kodu

Daha sonrasında 2 model oluşturulmuştur. Birinci ve ideal model olan Entropi değerlerine göre anahtar kare çıkaran Entropi modeli ilk olarak Entropi farklarını hesaplayan bir definition yazılmıştır.

```

# Calculate Average Entropy Difference for a chromosome.
def getEntropy( KF ):
    entropy_sum = 0
    for i in range(1, TOTAL_KEY_FRAMES - 1):
        while True:
            try:
                im1 = cv2.imread(READ_LOCATION + str(KF[i]) + ".jpg",0)
                im2 = cv2.imread(READ_LOCATION + str(KF[i+1]) + ".jpg",0)
                entropy_sum += abs(entropy(im1) - entropy(im2))
            except:
                print (i, KF, KF[i], KF[i+1])
                continue
            break
    return entropy_sum/(TOTAL_KEY_FRAMES - 1)

```

Figure 4: Entropi Değeri Hesaplama

Daha sonrasında Entropi değerlerine göre ebeyvenleri oluşturan bir kod yazılmıştır. Bu kodun amacı diferansiyel evrim algoritmasında işleme girecek olan karelerin seçilmesi ve daha sonra mutasyon,çaprazlama ve seleksiyon yapılacak bireylerin belirlenmesi için kullanılmıştır.

```

# INITIALISATION : Generates population NP of 10 parent vectors (and Average Entropy Differences).
def initialize_NP():
    for i in range(NUMBER_OF_NP_CANDIDATES):
        NP.append(sorted(random.sample(range(1, MAX_NUMBER_OF_FRAMES+1), TOTAL_KEY_FRAMES)))
        NP[-1].append(getEntropy(NP[-1]))
    print (NP[-1])

```


Figure 5: Ebeveyn Oluşturma Aşaması

Ebeveynlerin oluşturulmasından sonra Diferansiyel Evrim Algoritmasının işlemleri başlanmaktadır. Mutasyon, çaprazlama ve seleksiyon işlemleri düzenli olarak yapılmakta ve iterasyon sayısı bitene kadar en ideal kare sonuçları alınana kadar devam etmektedir.

```
# MUTATION
def mutation(parent):
    R = random.sample(NP,2)
    global MV
    MV[:] = []
    MV_value = 0
    print (NP[parent])
    for i in range(TOTAL_KEY_FRAMES):
        MV_value = int(NP[parent][i] + F*(R[0][i] - R[1][i]))
        if(MV_value < 1):
            MV.append(1)
        elif(MV_value > MAX_NUMBER_OF_FRAMES):
            MV.append(MAX_NUMBER_OF_FRAMES)
        else:
            MV.append(MV_value)
    MV.sort()
    MV.append(getEntropy(MV))

# CROSSOVER (uniform crossover with Cr = 0.6).
def crossover(parent, mutant):
    print ("mutant: ", mutant)
    print ("parent: ", parent)
    for j in range(TOTAL_KEY_FRAMES) :
        if(random.uniform(0,1) < Cr) :
            TV.append(mutant[j])
        else:
            TV.append(parent[j])
    TV.sort()
    TV.append(getEntropy(TV))
    print ("TV : ", TV)

# SELECTION : Selects offspring / parent based on higher Entropy diff. value.
def selection(parent, trail_vector):
    if(trail_vector[-1] > parent[-1]):
        parent[:] = trail_vector
        print ("yes", parent)
    else:
        print ("no")
```

Figure 6: Diferansiyel Evrim Aşamaları

Diferansiyel evrim algoritmasında ki mutasyon, çaprazlama ve seleksiyon işlemleri her yapıldığında en ideal ebeveyni seçip tekrar işleme sokmak gerekmektedir. Bu sayede bir sonra ki iterasyonlarda alınan sonuçlar her seferinde daha ideal ebeveynler ile yapılmaktadır. Bu sebep ile "best parent" adı altında bir definasyon yazılmıştır ve her işlemten sonra ideal ebeveynler seçilmektedir.

```
# bestParent returns the parent with then maximum Entropy diff. value.
def bestParent(population):
    Max_Entropy_value = population[0][-1]
    Best_Parent_Index = population[0]
    for parent in population:
        if (parent[-1] > Max_Entropy_value):
            Max_Entropy_value = parent[-1]
            Best_Parent_Index = parent
    return Best_Parent_Index
```

Figure 7: İdeal Ebeveyn Seçme Definasyonu

2. modelde farklı olarak bir definasyon yazılmıştır. Bu definasyonda ortalama öklidyen mesafesi hesaplanmaktadır. Bu sayede incelenen videoda aranan anahtar karelerden birbirine en yakın olanlar mesafe olarak seçilmektedir.

```

# Calculate AED for a chromosome.
def getAED( KF ):
    ED_sum = 0
    for i in range(1, TOTAL_KEY_FRAMES - 1):
        while True:
            try:
                im1 = cv2.imread(READ_LOCATION + str(KF[i]) + ".jpg",0)
                im2 = cv2.imread(READ_LOCATION + str(KF[i+1]) + ".jpg",0)
                ED_sum += cv2.norm(im1, im2, cv2.NORM_L2)
            except:
                print (i, KF, KF[i], KF[i+1])
                continue
            break
    return ED_sum/(TOTAL_KEY_FRAMES - 1)

```

Figure 8: Ortalama Öklid Mesafesi Hesaplama

2. Modelin kalan aşamalarında yine diferansiyel evrim algoritması kullanılmıştır ve optimizasyonda aynı yöntem kullanılmıştır.

5 Sonuç

Projemizde ilk olarak anahtar kare çıkarımı konusunda araştırmalar yapılmıştır. Bu araştırmalar sonucunda video tarama konusunda basit ama etkili bir yöntem olduğuna karar verilmiştir. Bu sebep ile bir çok çalışma yapılan bir alan haline geldiği fark edilmiştir.

İlk olarak proje veri olarak videolaran alınmış kareler bulunmamaktaydı. Bu sebep ile yazılan "Frame-Oluşturma" dosyası yazılmıştır. Bu lokal kameradan bir video çekerek 300 video karesi yakalamaktadır. Daha sonrasında 2 model oluşturulmuştur. Bu modellerden birincisi Öklid uzaklık mesafesi değerlerine göre anahtar kareleri seçmekteydi. Bu kareleri seçtikten sonra ebeveyn seçilen bu kareler üzerinde Diferansiyel Evrim algoritması ile optimizasyon işlemi yapılmaktadır. Bu işlemlerden sonra en ideal kareler seçilmektedir. İkinci modelimizde entropi değerlerine göre optimizasyon yapılacak kareler seçilmektedir. Bu kareler seçildikten sonra tekrardan aynı optimizasyon yöntemi ile ideal kareler seçilmektedir. Sonuç olarak modellerden seçilen kareler bir GIF dosyası içine atılarak hen GIF'leri oluşturulmuş hemde ideal kareler seçilmiş bulunmaktadır.

References

- [1] Guozhu Liu, and Junming Zhao, Key Frame Extraction from MPEG Video Stream, Information Processing (ISIP), Third International Symposium, Pages 423 – 427, October 2010
- [2] Damian Borth, Adrian Ulges, Christian Schulze, Thomas M. Breuel, Keyframe Extraction for Video Tagging and Summarization, Proceedings of Informatiktage, 2008.
- [3] Markos Mentzelopoulos and Alexandra Psarrou, Key-Frame Extraction Algorithm using Entropy Difference, Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, October 2004.
- [4] T. Liu, H.-J. Zhang, and F. Qi, "A novel video key-frame-extraction algorithm based on perceived motion energy model," IEEE Trans.Circuits Syst. Video Technol., vol. 13, no. 10, pp. 1006–1013, Oct.2003.an, M. Khan, W. Wu, A. Ullah, S. W. Baik, Multi-grade brain tumor classification using deep CNN with extensive data augmentation, J. Comput. Sci., 30 (2019), 174–182.

- [5] R. Hammoud and R. Mohr, "probabilistic framework of selecting effective key frames for video browsing and indexing," in International workshop on Real-Time Image Sequence Analysis, 2000.
- [6] Xiaomu Song, Guoliang Fan, "Joint Key-Frame Extraction and Object-Based Video Segmentation," wacv-motion, vol. 2, pp.126-131, IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) - Volume 2, 2005.