МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Лабораторные работы**

**по курсу «Цифровое видео»**

Выполнила: Алексюнина Ю.В

Группа: М8О-407

Преподаватель: А.В. Крапивенко

Москва, 2021

**Методики объективной оценки потерь качества.**

1. **Цель**: Получение основных навыков вычисления объективных метрик качества видео.
2. **Индивидуальное задание:**
3. Для полученной на предыдущей ЛР пары роликов с диалогом двух персонажей, произвести вычисления объективных метрик по сумме всего времени воспроизведения:
   1. изменения разности матожидания;
   2. изменения метрики sup;
   3. изменения метрики PSNR;
   4. изменения метрики SSIM.
4. В отчете привести:
   1. пример, содержащий линейную разность одного из характерных кадров тестируемого видео (рекомендуется её инвертировать и визуально усилить);
   2. графики изменения указанных выше метрик с течением времени;
   3. вывод о предположительном попадании субъективных экспертных оценок в доверительные интервалы качества метрик PSNR и SSIM.
5. **ПО:** MSU Video Quality Measurement Tool, Jupyter Notebook
6. **Ход работы:**
7. Визуально усиленная линейная разность одного из характерных кадров тестируемого видео (frame 1482):



Значения метрик на выбранном кадре:

PSNR – 14.3

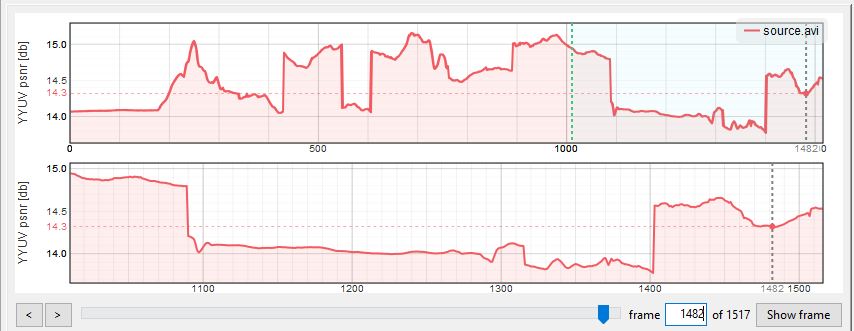
SSIM – 0.683

MSAD – 22.0

sup – 230

1. Графики изменения метрик psnr, ssim, msad, sup с течением времени:

* PSNR



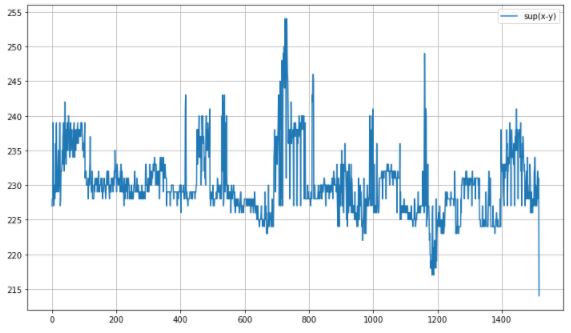
* SSIM

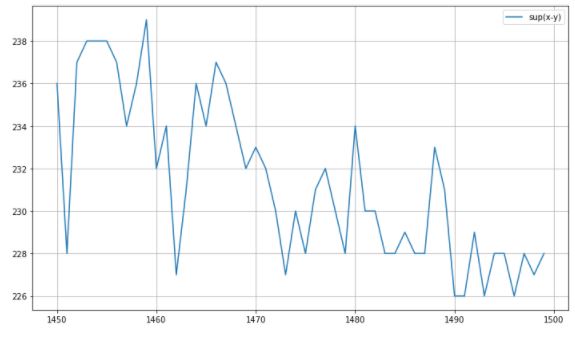


* MSAD



* sup(x-y):





Код программы для вычисления sup:

import cv2

vidcap = cv2.VideoCapture('D:/source.avi')

success,image = vidcap.read()

vidcap2 = cv2.VideoCapture('D:jpeg2000.mxf')

success2,image2 = vidcap2.read()

count = 0

count2 = 0

while success or success2:

if success:

cv2.imwrite("D:/frames1/frame%d.jpg" % count, image) # save frame as JPEG file

success,image = vidcap.read()

print('Read a new frame: ', success)

count += 1

if success2:

cv2.imwrite("D:/frames2/frame%d.jpg" % count, image2) # save frame as JPEG file

success2,image2 = vidcap2.read()

count2 += 1

import glob

images = [cv2.imread(file) for file in glob.glob("D:/frames1/\*.jpg")]

images2 = [cv2.imread(file) for file in glob.glob("D:/frames2/\*.jpg")]

images\_yuv = images

images\_yuv2 = images2

for i in range(0, count-1):

images\_yuv[i] = cv2.cvtColor(images[i], cv2.COLOR\_BGR2YUV) # images[i].convert('YCbCr')

for i in range(0, count2-1):

images\_yuv2[i] = cv2.cvtColor(images2[i], cv2.COLOR\_BGR2YUV) # images[i].convert('YCbCr')

diff = [] # массив модулей разностей

diff\_t = []

sup = []

for i in range(0, len(images\_yuv)): # for everyimage in images\_yuv, images\_yuv2:

Y\_i, U\_i, V\_i = cv2.split(images\_yuv[i])

Y\_i2, U\_i2, V\_i2 = cv2.split(images\_yuv2[i])

for Y, Y2 in zip(Y\_i,Y\_i2):

for val, val2 in zip(Y, Y2):

diff\_t.append(abs(int(val) - int(val2)))

diff.append(diff\_t)

diff\_t = []

max\_Y = 0

for yy in diff:

if max(yy) > max\_Y:

max\_Y = max(yy)

sup.append(max\_Y)

diff = []

#строим весь график

import matplotlib.pyplot as plt

x = []

for i in range(0, 1517):

x.append(i)

plt.figure(figsize=(12, 7))

plt.plot(x, sup, label="sup(x-y)")

plt.legend()

plt.grid(True)

# масштабируем

x\_temp = []

y\_temp = []

for i in range(1450, 1500):

x\_temp.append(x[i])

y\_temp.append(sup[i])

plt.figure(figsize=(12, 7))

plt.plot(x\_temp, y\_temp, label="sup(x-y)")

plt.legend()

plt.grid(True)

1. Выводы:

* среднее значение PSNR < 20 на протяжении всего ролика
* среднее значение SSIM ~ 0.7 на протяжении всего ролика

Можно сделать вывод о том, что качество сжатого ролика заметно хуже, чем исходного, тем не менее, показатели ssim на хорошем уровне, так что, возможно, что субъективных экспертные оценки попадут в доверительные интервалы качества метрик PSNR и SSIM.