ДАМЫ и ГОСПОДА,

СЕГОДНЯ **МЫ** ПРЕССУЕМ Гослинга

А ЕСТЬ ли ему до этого дело?

В качестве изображения для задания 1 возьмем изображение Гослинга с титульного листа

Для простоты рассмотрим лишь часть кода, которая выполняет основную логику программы

```
1  def compress_grayscale_img(img: Image, compr_perc: int):
2      gray_img_arr = np.asarray(img.convert("L"))
3      U, S, V = np.linalg.svd(gray_img_arr)
4
5      dim_S = S.shape[0]
6      compr_val = dim_S - int(dim_S * (compr_perc / 100))
7
8      return Image.fromarray(
9          np.matrix(U[:, :compr_val])
10         * np.diag(S[:compr_val])
11         * np.matrix(V[:compr_val, :])
12     )
```

~$ На вход функции мы подаем изображение и процент сжатия

~$ На 2 строчке изображение преобразуется в черно-белое и конвертируется в список

~$ На 3 строчке наш список превращается в SVD разложение

~$ На 6 строчке вычисляется количество векторов k, которые мы оставим в разложении (S в данном случае – одномерный вектор, а U и V – квадратные матрицы)

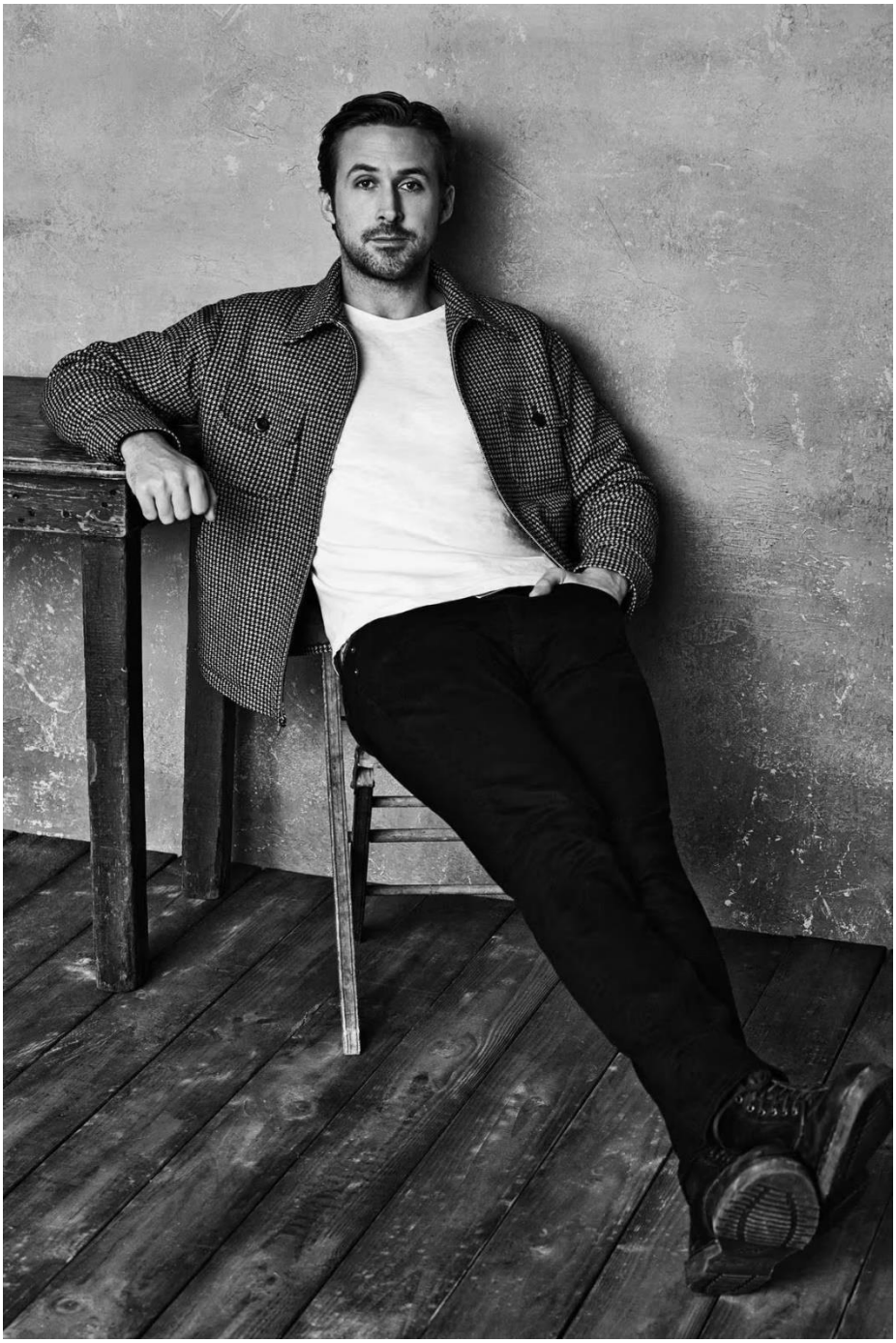
~$ На 8 строчке «списки» U, S и V срезаются до значения переменной compr_val, перемножаются и конвертируются обратно в изображение

Пример использования функции:

```
1  img_path = png_imgs[img_to_open - 1]
2  img = img_utils.read_image(img_path)
3  compr_img = img_utils.compress_grayscale_img(img, compr_perc)
```

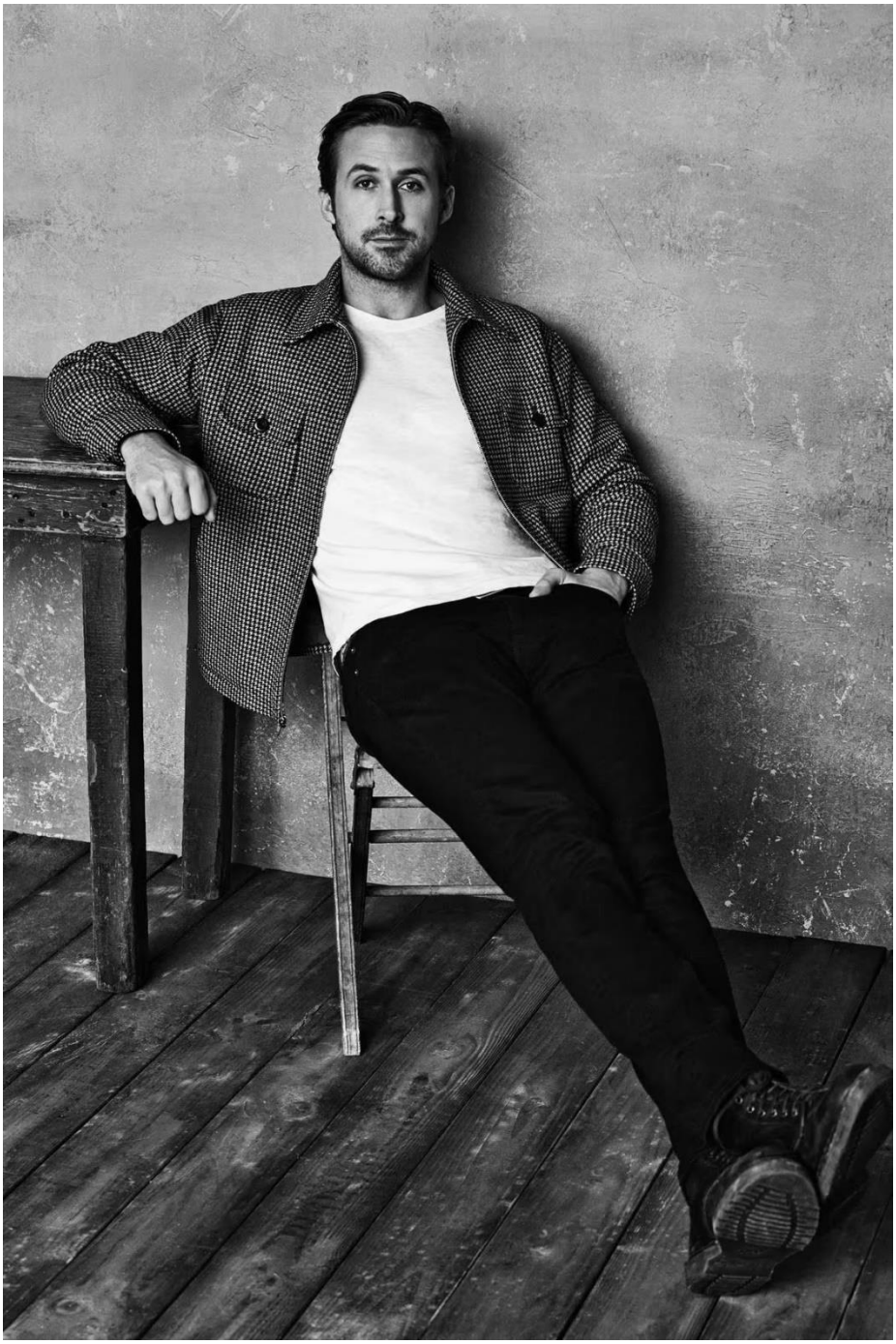
Теперь перейдем к сжатию Гослинга! Атрибут shape матрицы сингулярных чисел равен 1173, это 100%

Пусть k=1056, что соответствует сжатию изображения на 10% (Оставили 1056 векст.). Получим следующий результат:



Кажется, Гослинг чувствует себя отлично. Он прекрасно различим, разве что фото теперь датируется парой сотен лет раньше, я думаю для него это не проблема

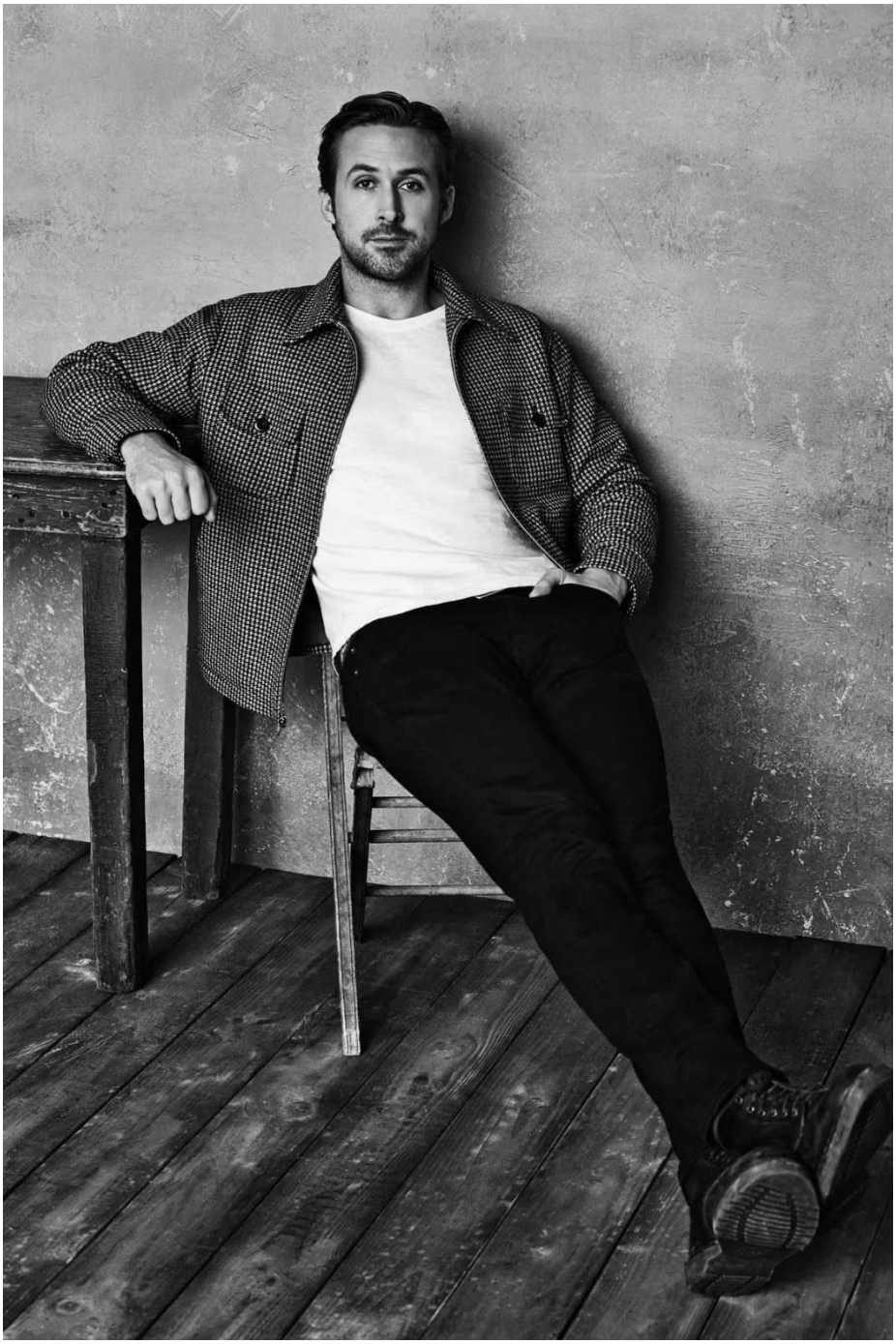
Пусть k=939, что соответствует 20% сжатия изображения



Все еще 10 Гослингов из 10. Заметно небольшое уменьшение веса изображения – оригинал весит 2.54 МБ, сжатое на 20% весит 2.21 Мб

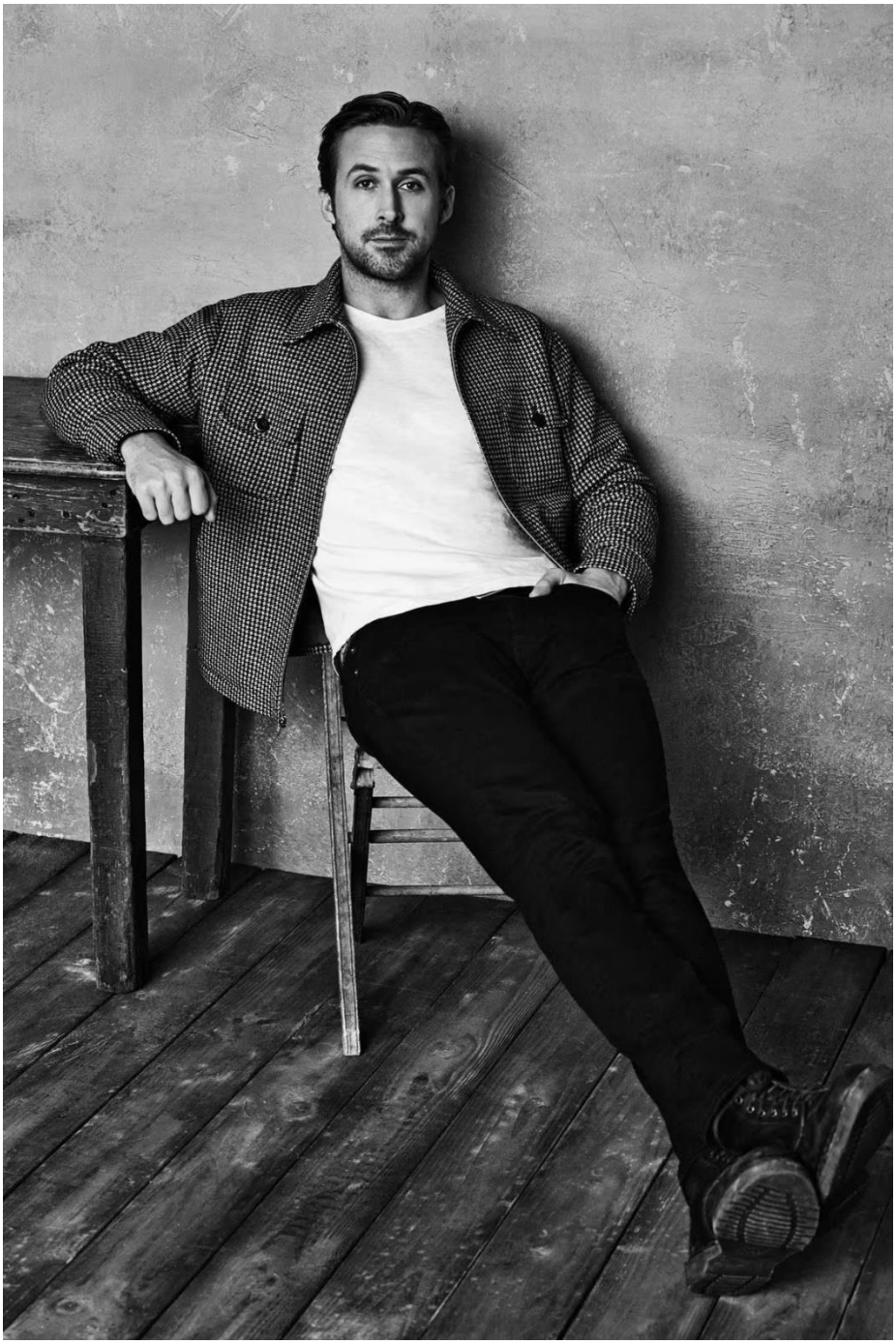
Для одного фото Гослинга и целой одной страницы не жаль

Пусть k=822, что соответствует 30% сжатия изображения



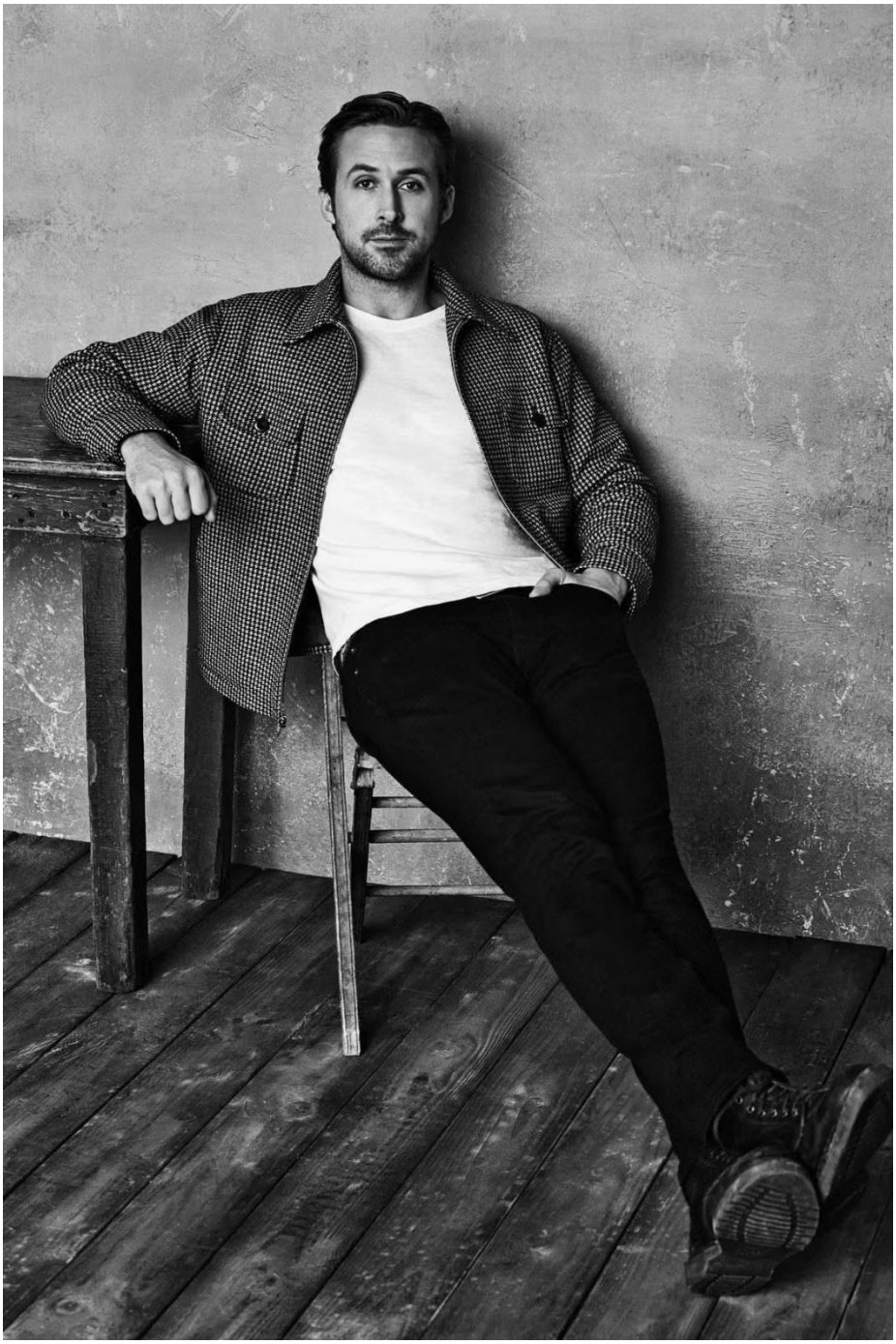
А что-то изменилось? Может быть, моя программа на самом деле не работает? Или это некая Гослингова сила, что не позволяет нам испортить его внешний вид? Само изображение стало весить 2.25 МБ. Это больше, чем в прошлый раз, причем само изображение занимает меньше оперативной памяти компьютера внутри программы

Пусть k=704, что соответствует 40% сжатия изображения



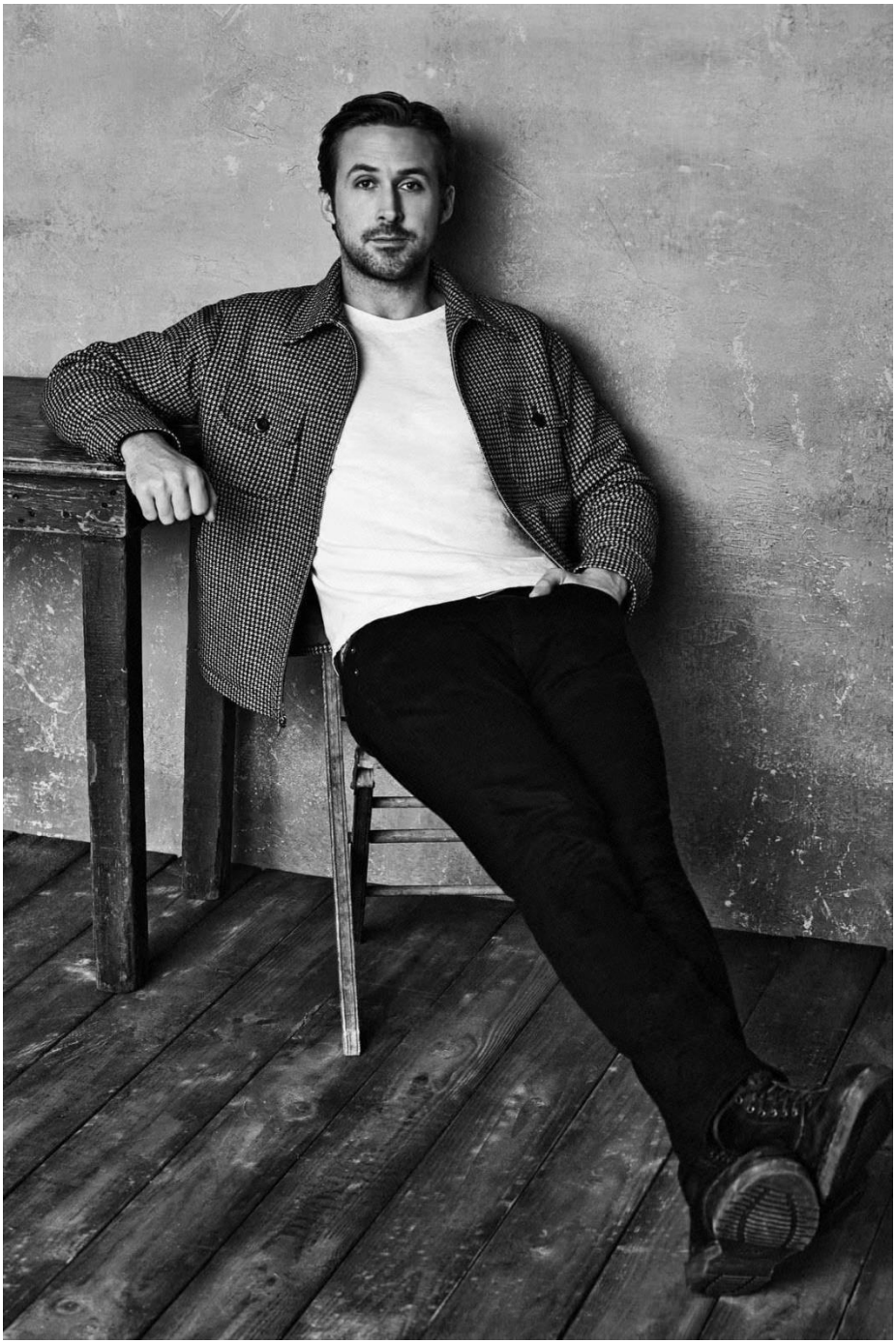
Это почти половина от оригинала. А изображение, хи-хи ха-ха, весит еще больше – 2.27 МБ. Пока как-то Гослинг не прессуется. Похоже, SVD для него расшифровывается как Sigma Valuable Defined

Пусть k=587, что соответствует 50% сжатия изображения



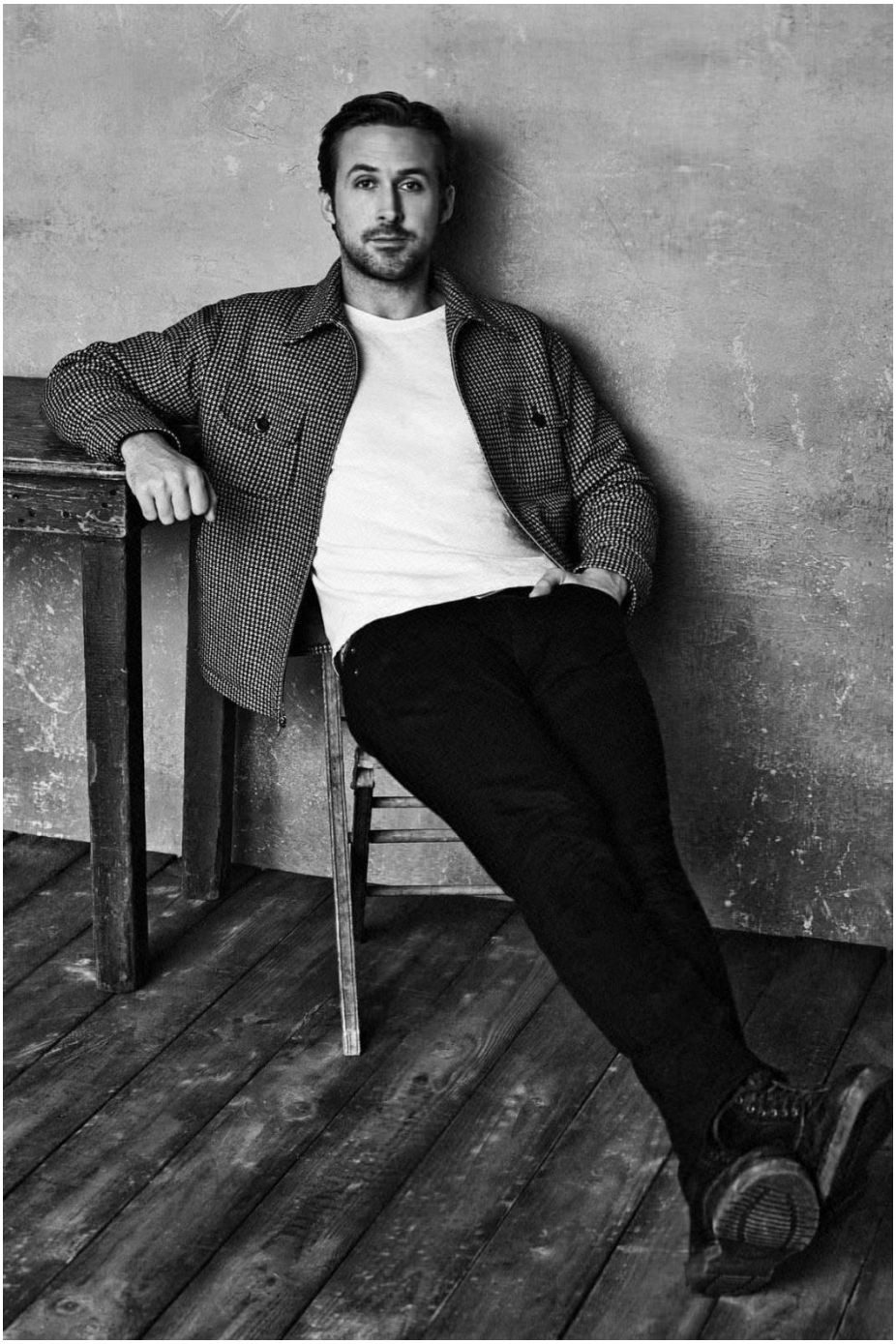
Победа! Наш Гослинг наконец-то похудел. На целый 0.01 МБ. Играем в игру «Найди 625 отличий». Похоже, я проиграл. Может вы что увидите? Половину съели, а по факту он все также хорош

Пусть k=470, что соответствует 60% сжатия изображения



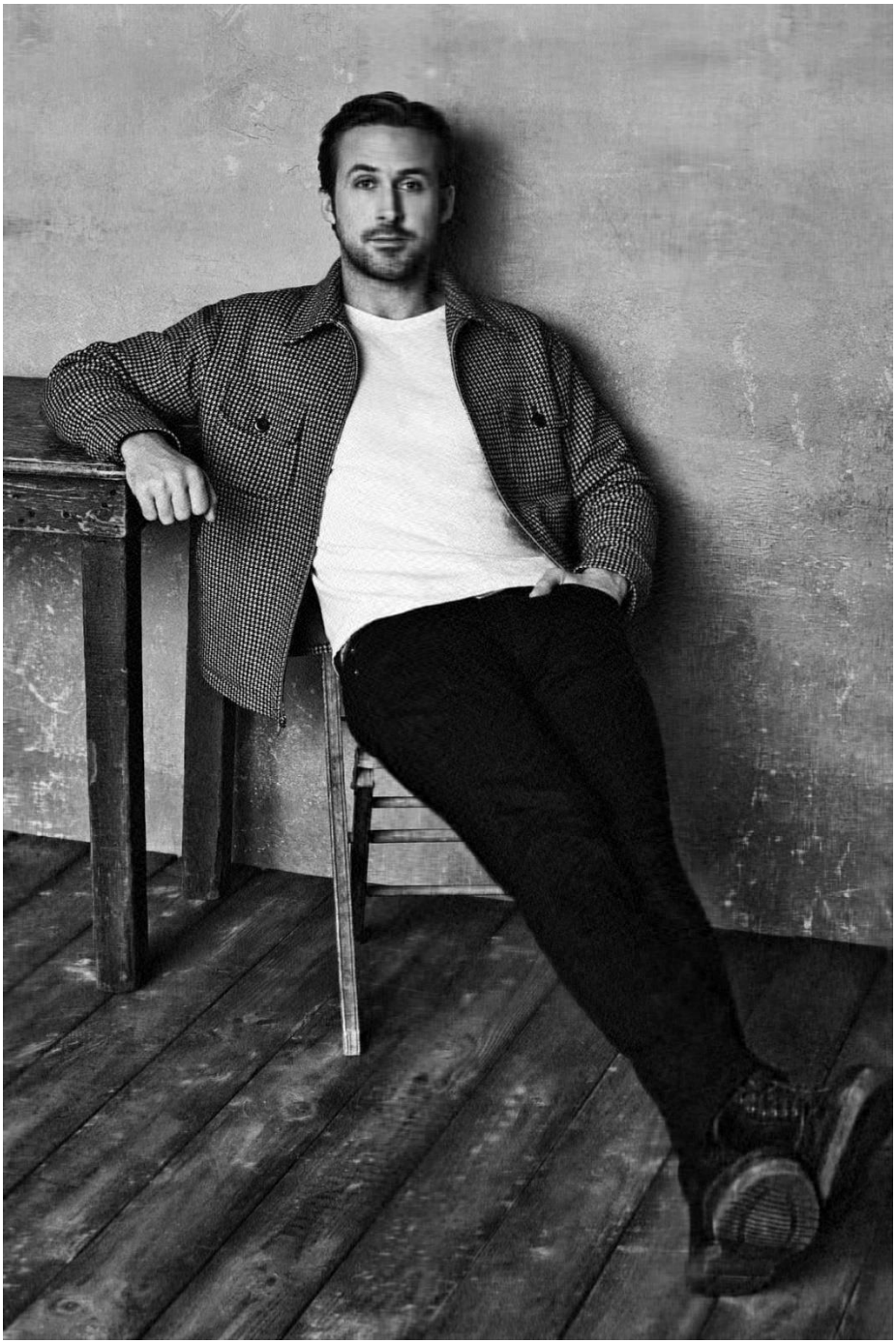
Да все равно ему. А вы представьте, я сижу, выдумываю и пишу этот текст, общаюсь с некоторым «собеседником» в каком-то там Word, а ведь это может остаться непрочтенным. Ладно, мне просто нужно было побольше текста для объема. И чтобы читающий не заскучал. Ты же не заскучал?

Пусть k=352, что соответствует 70% сжатия изображения



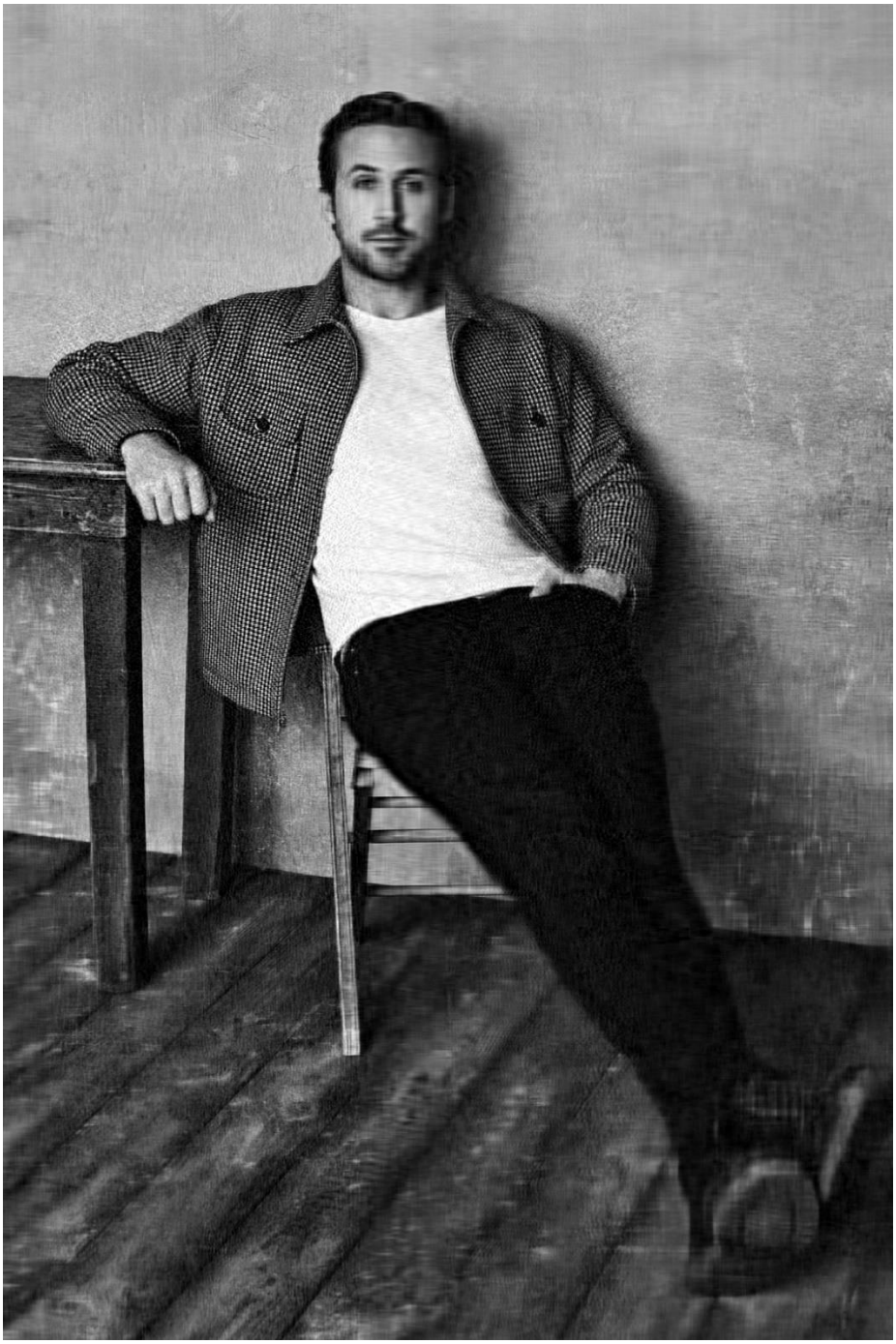
Вы, наверное, думали, что я 6 раз подряд вставил одну и ту же картинку? Это был бы умный ход, но вот тут наш Гослинг начал сдавать. Посмотрите на правую часть его лица – оно уже немного смазано, но нужно хорошо вглядеться, чтобы это заметить. В прочем это не мешает ему все также сидеть на своем стуле в позе «Мне дела нет»

Пусть k=235, что соответствует 80% сжатия изображения



А вот тут уже заметно, что лицо начинает съезжать и мылиться. Вы, наверное, соскучились по мегабайтовой статистике? Сейчас наш Гослинг весит 2.04 МБ. Изображение хоть и стало хуже, но на аватарку еще ставить можно

Пусть k=118, что соответствует 90% сжатия изображения



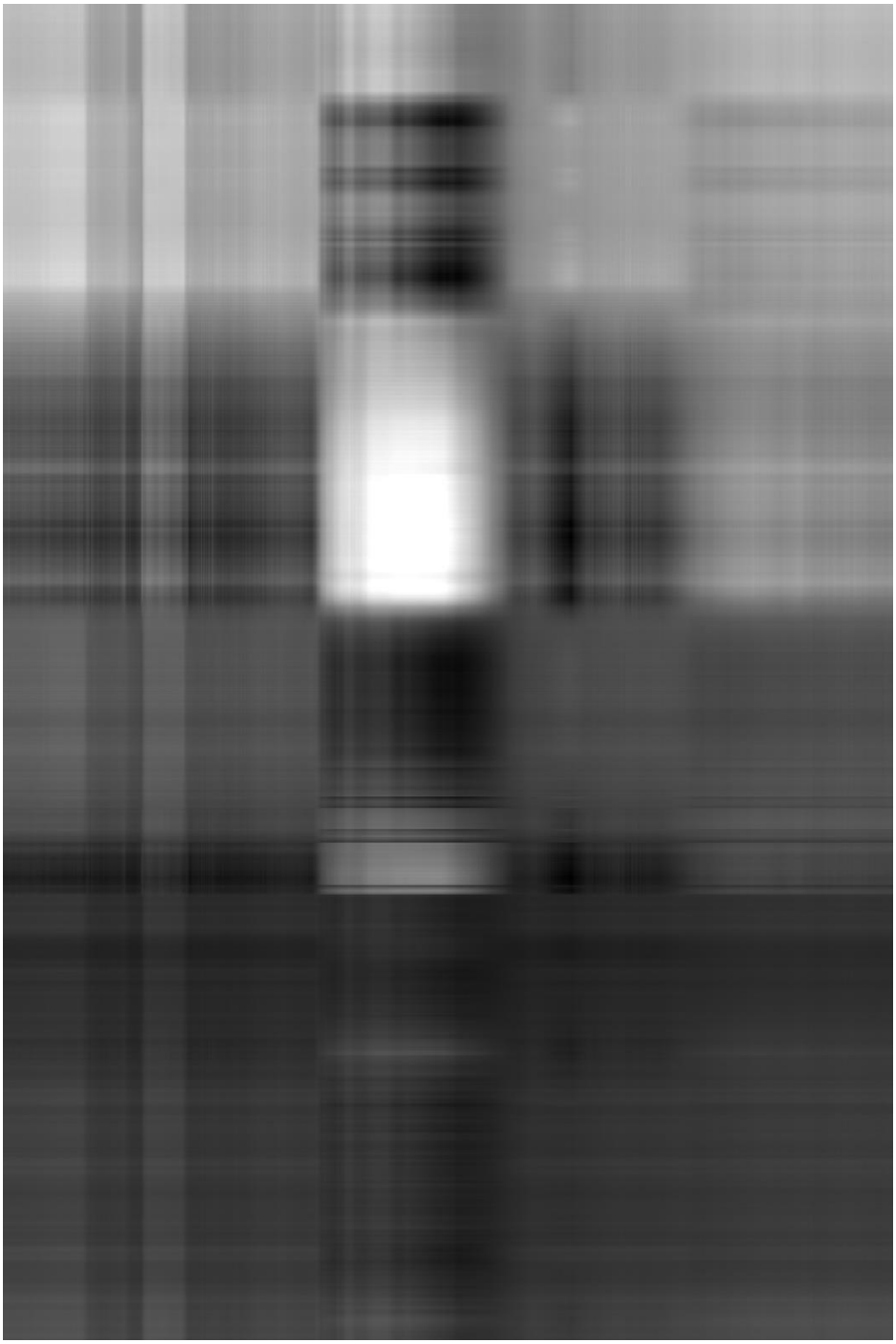
А вот тут уже не все равно. Ботинки, лицо и края изображения уже заметно замылились, однако сам Гослинг еще более чем различим. С каждым сжатием фото будто стареет на некоторое количество лет. Вес, кстати, тоже упал – уже 1.82 МБ

Пусть k=59, что соответствует 95% сжатия изображения

Это уже много, изображение будто начало расходиться по швам. Похоже, мы удалили довольно значимые для картинки вектора, но Гослингу то вайолентово. Он все еще различим, пускай и выглядит так себе. Вес изображения 1.59 МБ

Ну а теперь…

Пусть k=2, что соответствует 99.9% сжатия изображения



Штош, вот он и проиграл моей программе. Аплодисменты, дорогие зрители, спектакль картинок подошел к концу (или нет). Гослинг сильно исхудал, весит всего 567 КБ. Если это чудо конечно можно назвать Гослингом. Скорее похоже на страшилку из Minecraft, аля Гослингрин или что-то типа того

Сделаем выводы

Чем больше значение параметра k, тем лучше выглядит изображение. Однако даже при низких значениях k изображение все еще выглядит хорошо, так как мы оставляем самые весомые вектора, соответствующие самым большим сингулярным числам

В памяти достаточно хранить 1/2 или 1/3 информации от исходной

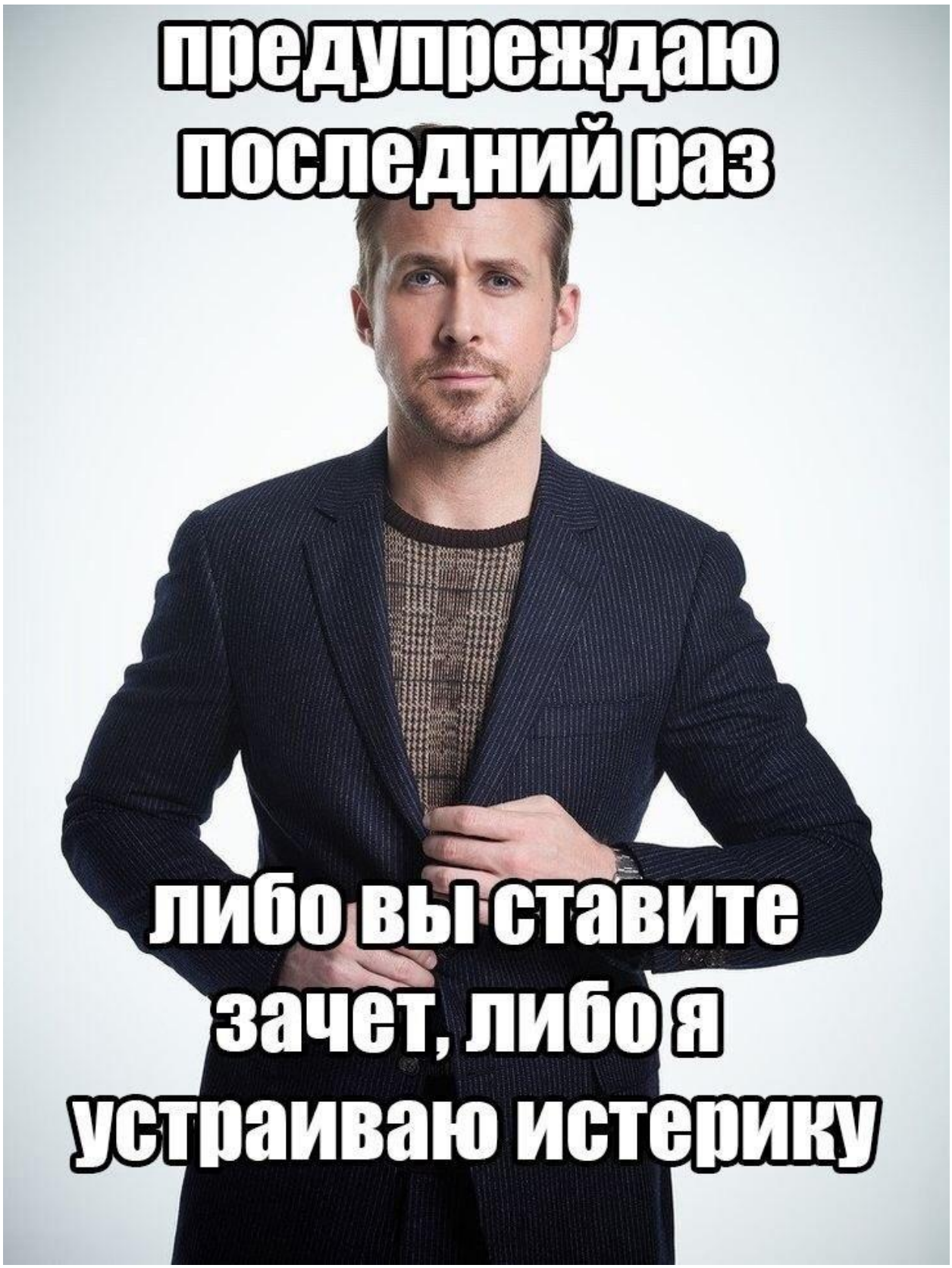
В моем случае, картинка имеет приемлемое качество при параметре k>=352. Плохое качество, но различимое изображение при 59<=k<352. Картинка становится совершенно непонятной при 1<=k<59

Степени сжатия в процентах относительно SVD разложения я уже написал выше

Оценим степени сжатия по размеру файла картинки:

1. 2.19/2.54 = 0.862 МБ
2. 2.21/2.54 = 0.87 МБ
3. 2.25/2.54 = 0.886 МБ
4. 2.27/2.54 = 0.894 МБ
5. 2.26/2.54 = 0.89 МБ
6. 2.23/2.54 = 0.878 МБ
7. 2.16/2.54 = 0.85 МБ
8. 2.04/2.54 = 0.803 МБ
9. 1.82/2.54 = 0.717 МБ
10. 1.59/2.54 = 0.626 МБ
11. 0.567/2.54 = 0.223 МБ

Вот вы и дошли до конца. Дальше ничего не будет


предупреждаю
последний раз
либо вы ставите
зачет, либо я
устраиваю истерику