

Алгоритм формирования датасета

Данный материал описывает алгоритм формирования датасета. Датасет представляет из себя изображения 8 на 8 пикселей, разделенных на 2 класса. Сами классы соответствуют двум схемам встраивания: класс 1 соответствует экспериментальной схеме встраивания 1 (см Изображение 1), класс 2 соответствует экспериментальной схеме встраивания 2 (см Изображение 2). В ходе формирования датасета будут использоваться такие атаки как «JPEG70» и «Увеличение контрастности», поскольку именно они в ходе экспериментов оказывали самые деструктивные воздействия на изображения.

DC	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

1- region

0 - region

Рисунок 1. Исходная схема встраивания 1.

DC	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

1- region

0 - region

Рисунок 2. Экспериментальная схема встраивания 2.

Алгоритм формирования датасета

Вход: Изображение I , пороговое значение ошибок τ_{max} .

Выход: Две папки Dir_1 и Dir_2 с изображениями 8 на 8 пикселей, соответствующие двум классам, папка Dir_{rand} . с изображениями 8 на 8 пикселей, которые не были определены ни к одному из классов.

1. Создаем 2 копии изображения I для каждого класса: $I_0^1, I_1^1, I_0^2, I_1^2$.
2. Встраиваем 0 и 1 в копии изображения I следующим образом:

- 2.1. В I_0^1 встраиваем 0, используя исходную схему.
- 2.2. В I_1^1 встраиваем 1, используя исходную схему.
- 2.3. В I_0^2 встраиваем 0, используя экспериментальную схему.
- 2.4. В I_1^2 встраиваем 1, используя экспериментальную схему.
3. Атакуем каждую копию, используя атаки «JPEG70» и «Увеличение контрастности», после чего получим 12 копий исходного изображения: 4 не атакованные копии ($I_0^1 \dots I_1^2$), 4 копии, искаженные атакой «JPEG70» ($(I_0^1)^{JP} \dots (I_1^2)^{JP}$), 4 копии, искаженные атакой «Увеличение контрастности» ($(I_0^1)^{CI} \dots (I_1^2)^{CI}$).
4. Делим все копии изображения I на блоки.
5. Блоки изображений каждого класса объединяются в группы в соответствии с их индексом. Всего групп блоков для каждого класса: 4096, в каждой группе по 6 блоков:
 - 5.1. Блок изображения со встроенным битом 0.
 - 5.2. Блок изображения со встроенным битом 1.
 - 5.3. Блок изображения (JPEG70) со встроенным битом 0.
 - 5.4. Блок изображения (JPEG70) со встроенным битом 1.
 - 5.5. Блок изображения (Увеличение контрастности) со встроенным битом 0.
 - 5.6. Блок изображения (Увеличение контрастности) со встроенным битом 1.
6. Для каждой группы двух классов извлекаем ранее встроенные биты и считаем количество ошибочно извлеченных битов: τ_1 — для изображений 1 класса τ_2 — для изображений второго класса.
7. Для блока M исходного изображения I класс определяется следующим образом:
Для $i = \overline{1, N}$, где N — количество блоков в изображении, выполним следующее:
 - 6.4. Если $\tau_1 < \tau_{max}$, то записать M[N] в Dir_1 .
 - 6.5. Если $\tau_2 \leq \tau_{max} \leq \tau_1$, то записать M[N] в Dir_2 .
 - 6.6. Иначе записать M[N] в Dir_{rand} .
8. Сохранение Dir_1 , Dir_2 и Dir_{rand} .
В случае перевеса одного из классов или большого количества изображений в папке Dir_{rand} следует изменить параметр τ_{max} .