

1. Анализ данных:

- Описание набора данных (размер, признаки, целевая переменная).

Размер – 42000x785

Каждая строка – изображение 28x28 пикселей (высота, ширина).

Каждый пиксель имеет соответствующее ему значение ([0:255]), указывающее на яркость или темноту этого пикселя (более большие числа означают более темный оттенок)

Обучающий набор данных (train.csv) содержит 785 столбцов.

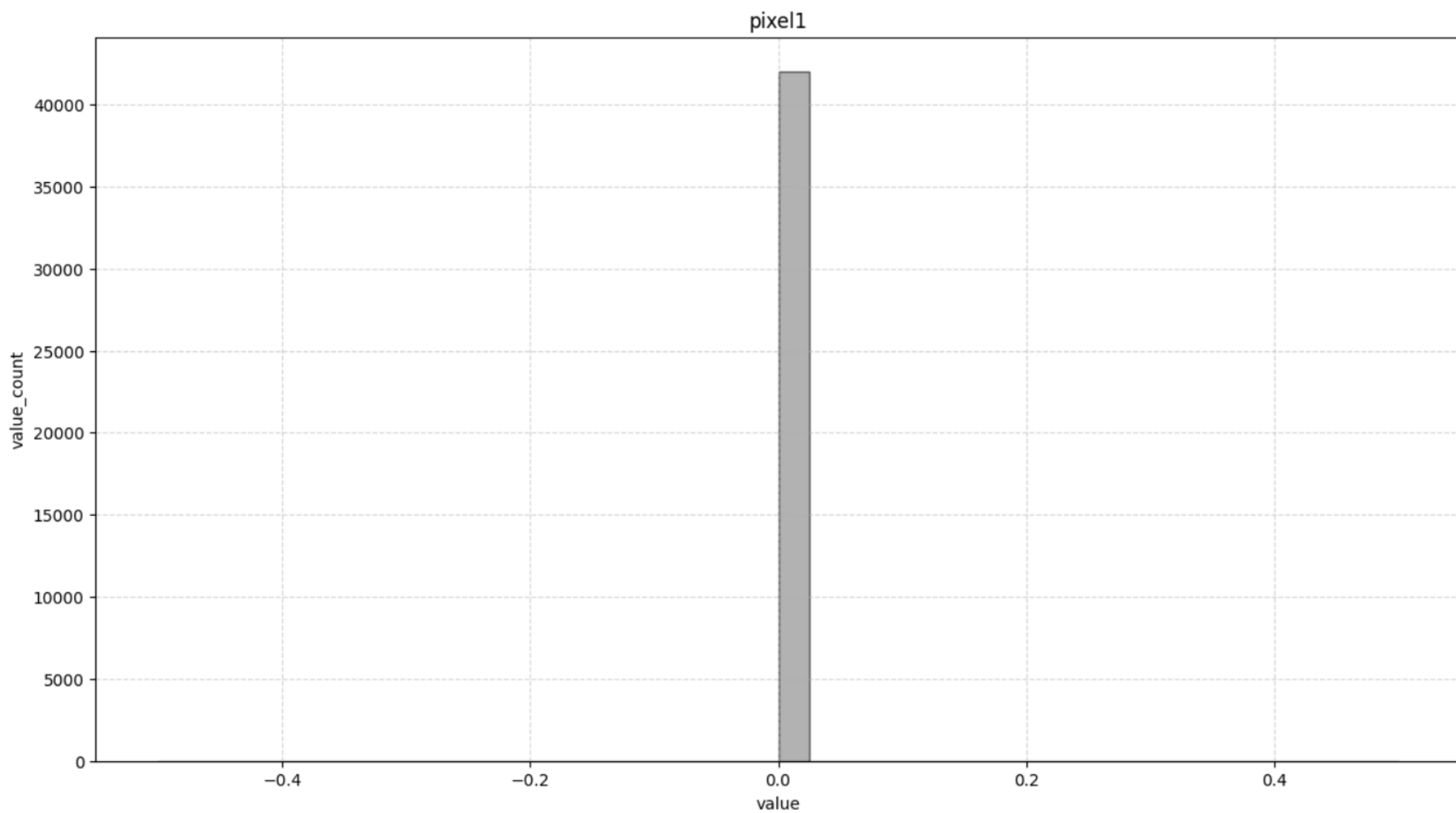
Целевая переменная – первый столбец 'label' – содержит целое число.

Признаки – столбцы с названием вида pixelx, где x — целое число от [0:783]. Содержат значения пикселей соответствующего изображения

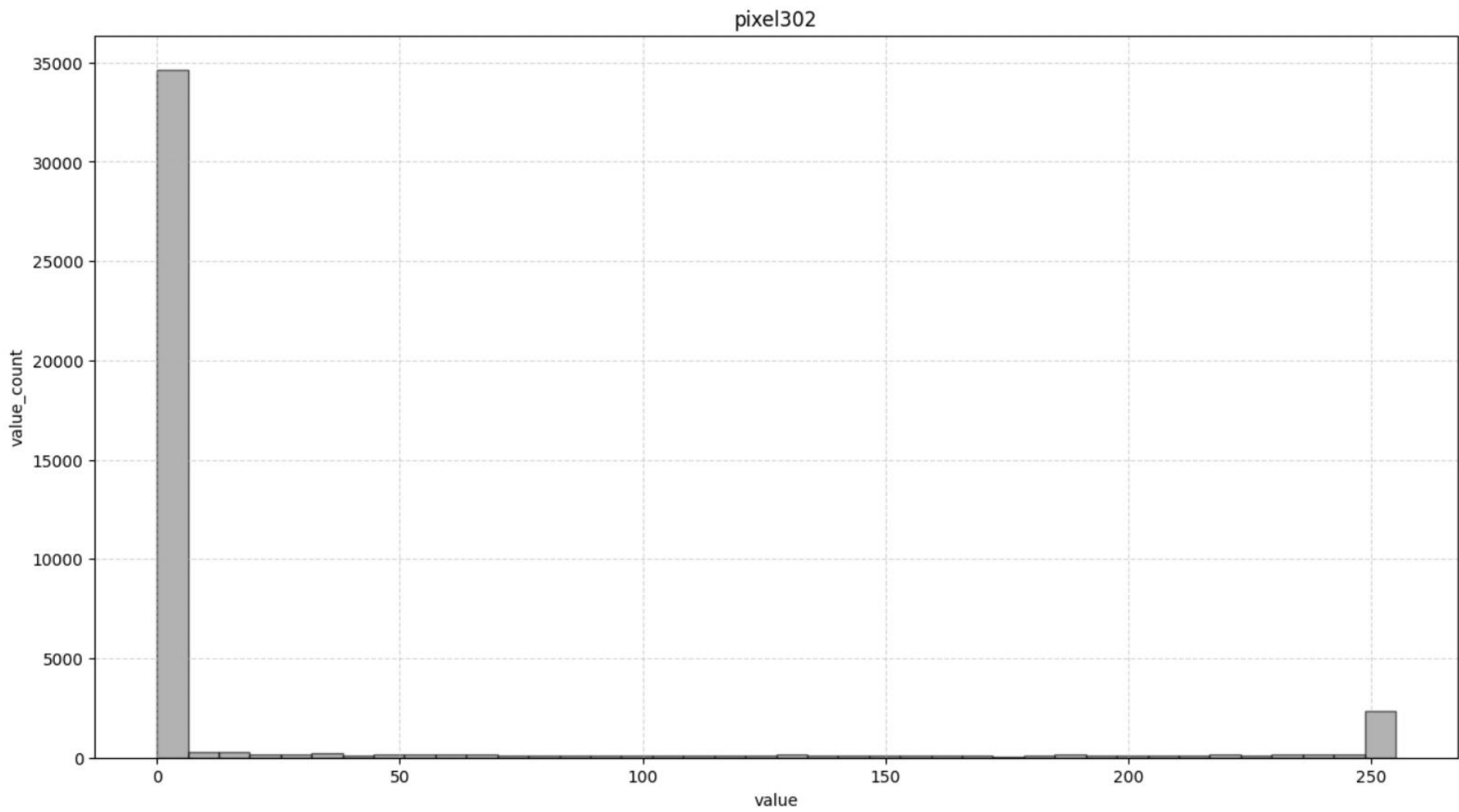
Тестовый набор данных (test.csv) аналогичен обучающему, за исключением того, что в нем отсутствует столбец label – таргет.

- Визуализация распределений (гистограммы, box-plot).

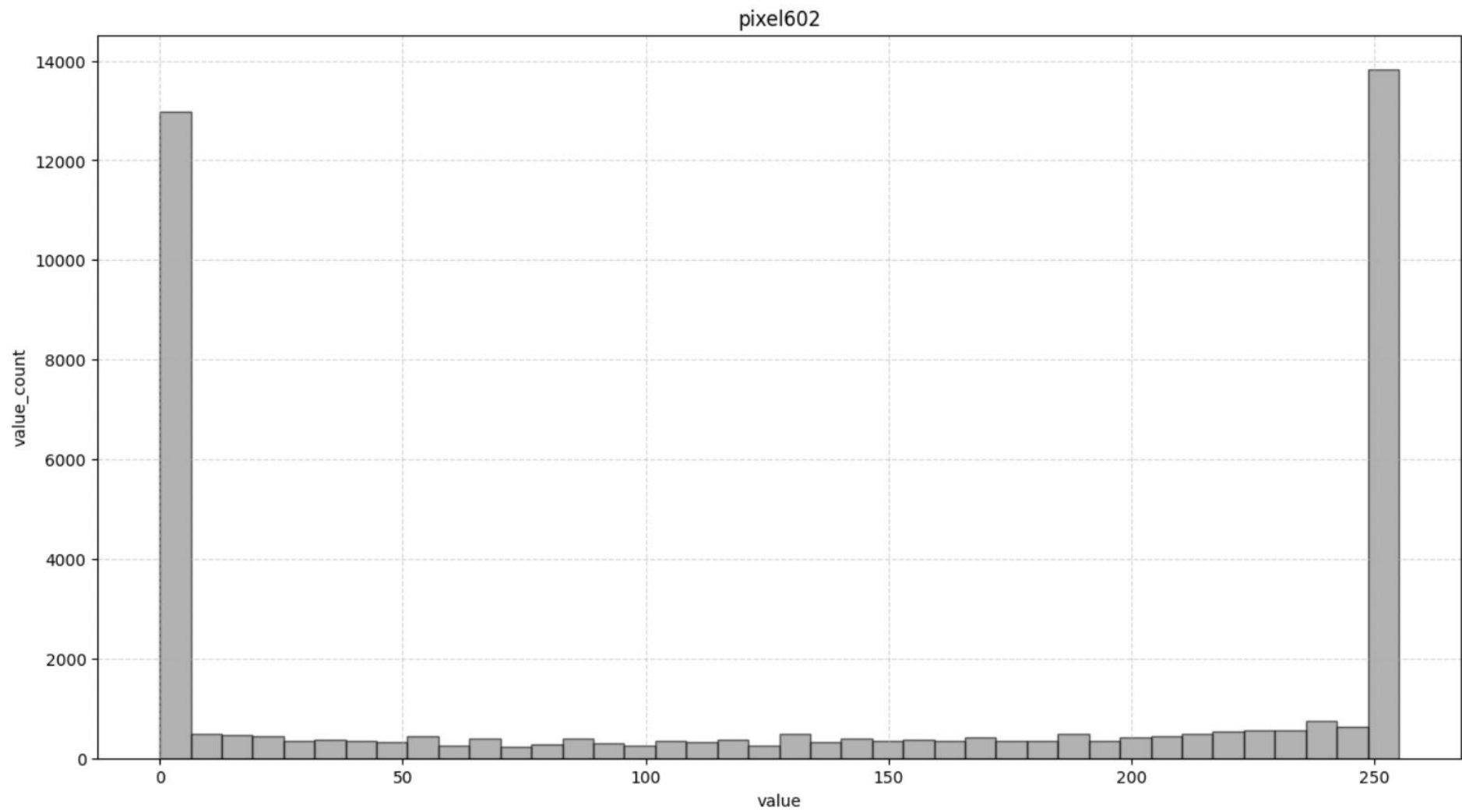
По одному признаку (рис.1-4):



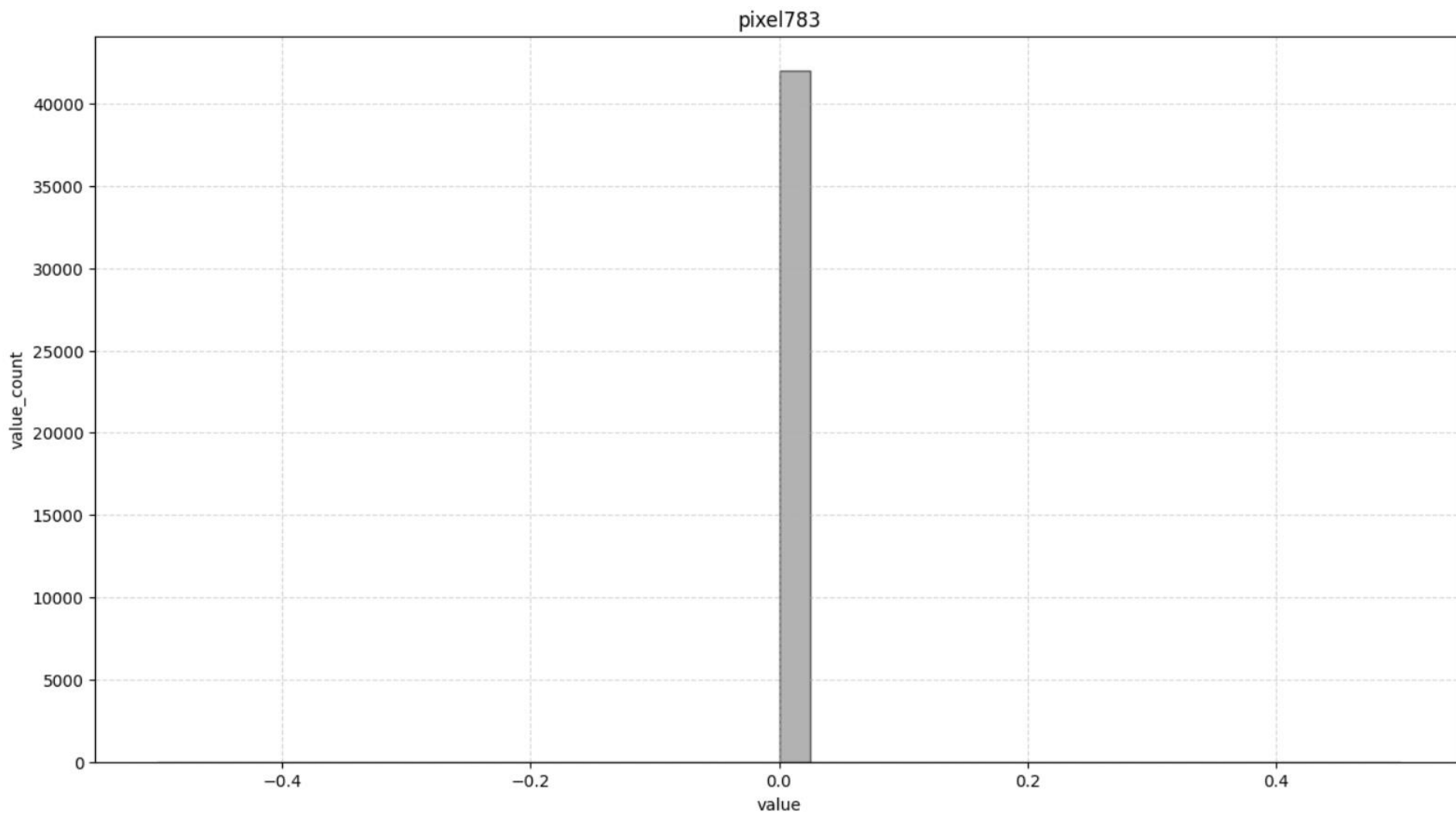
(рис.1)



(рис.2)



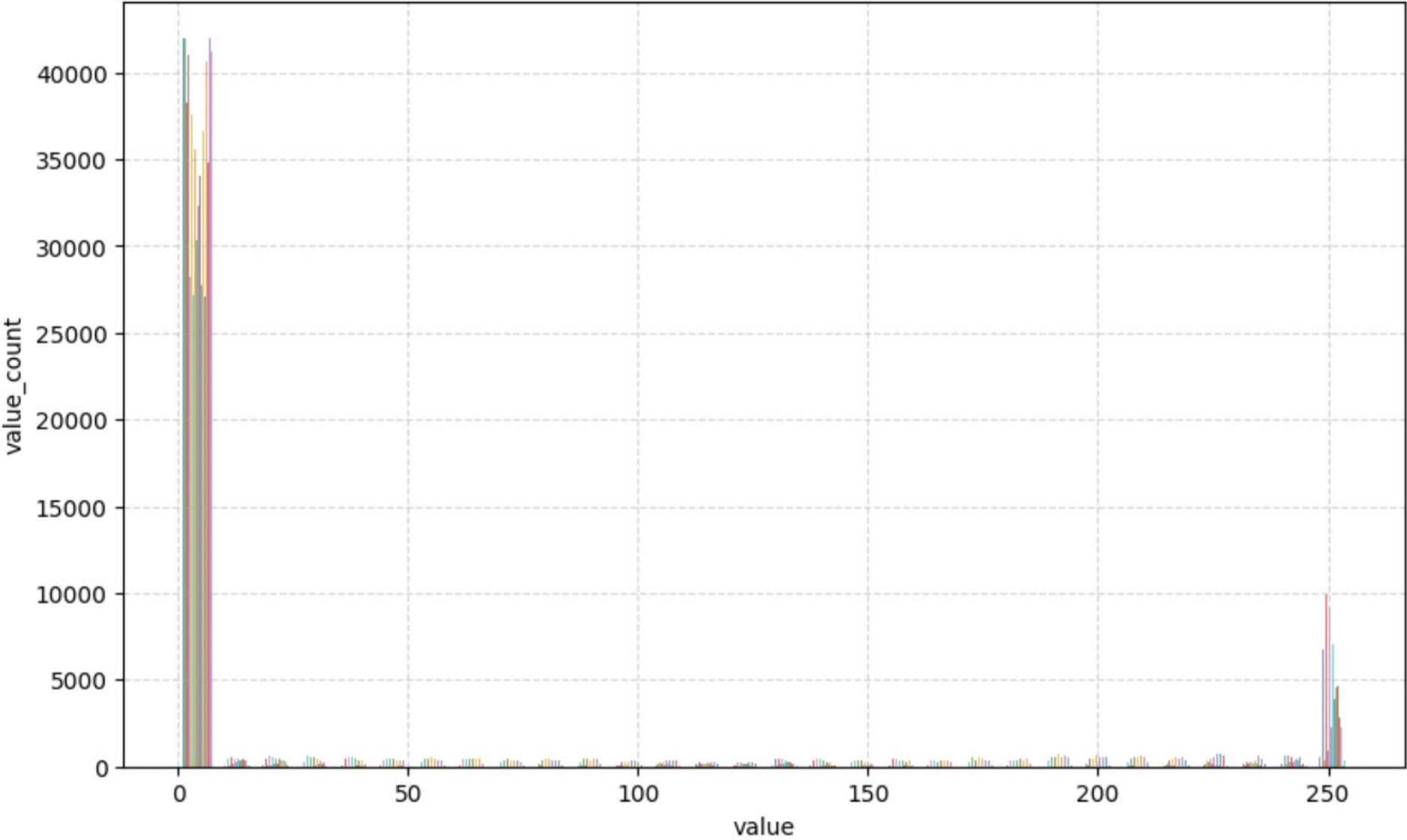
(рис.3)



(рис.4)

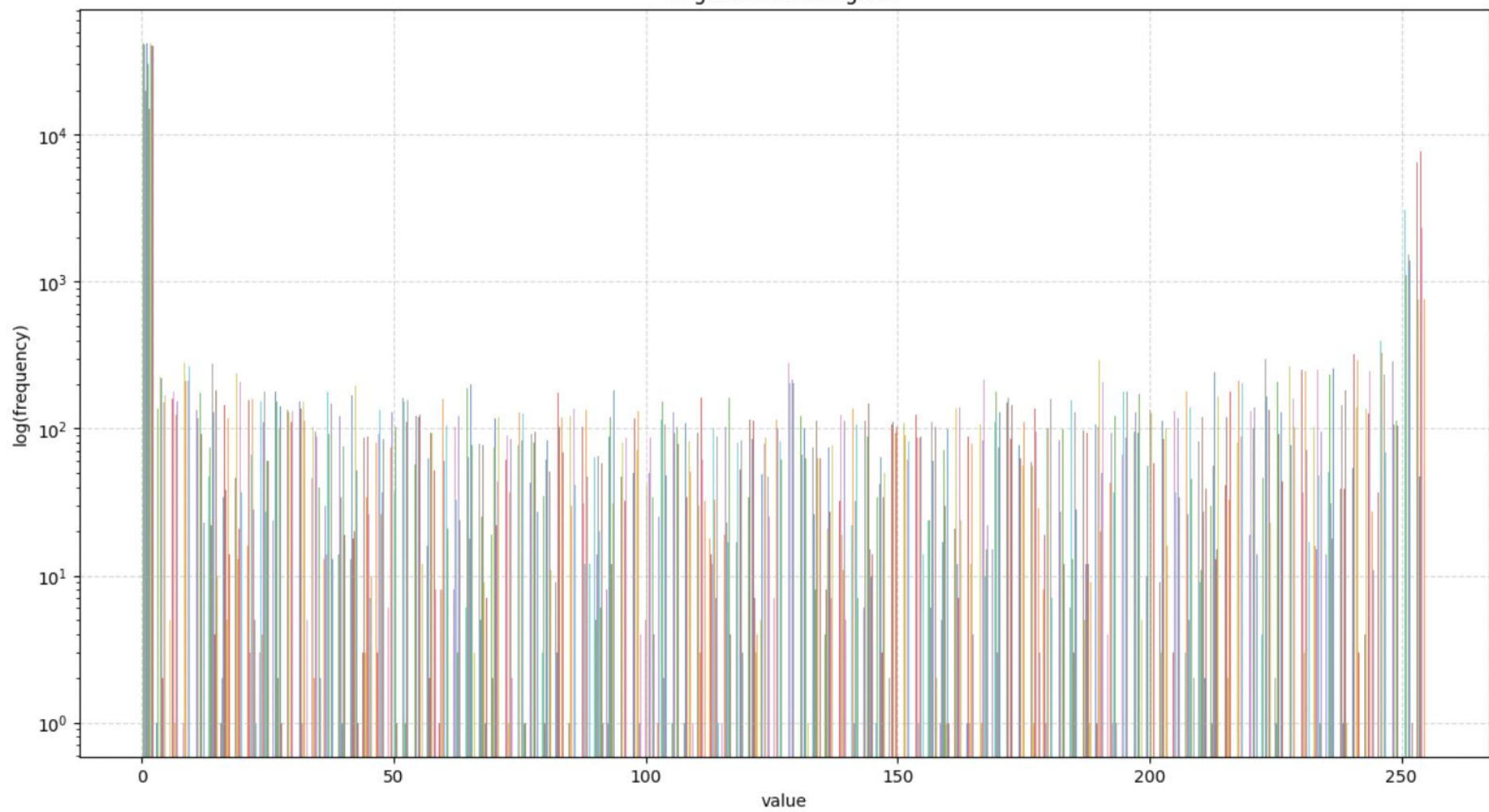
По всем признакам (рис.5-7):

All features histogram



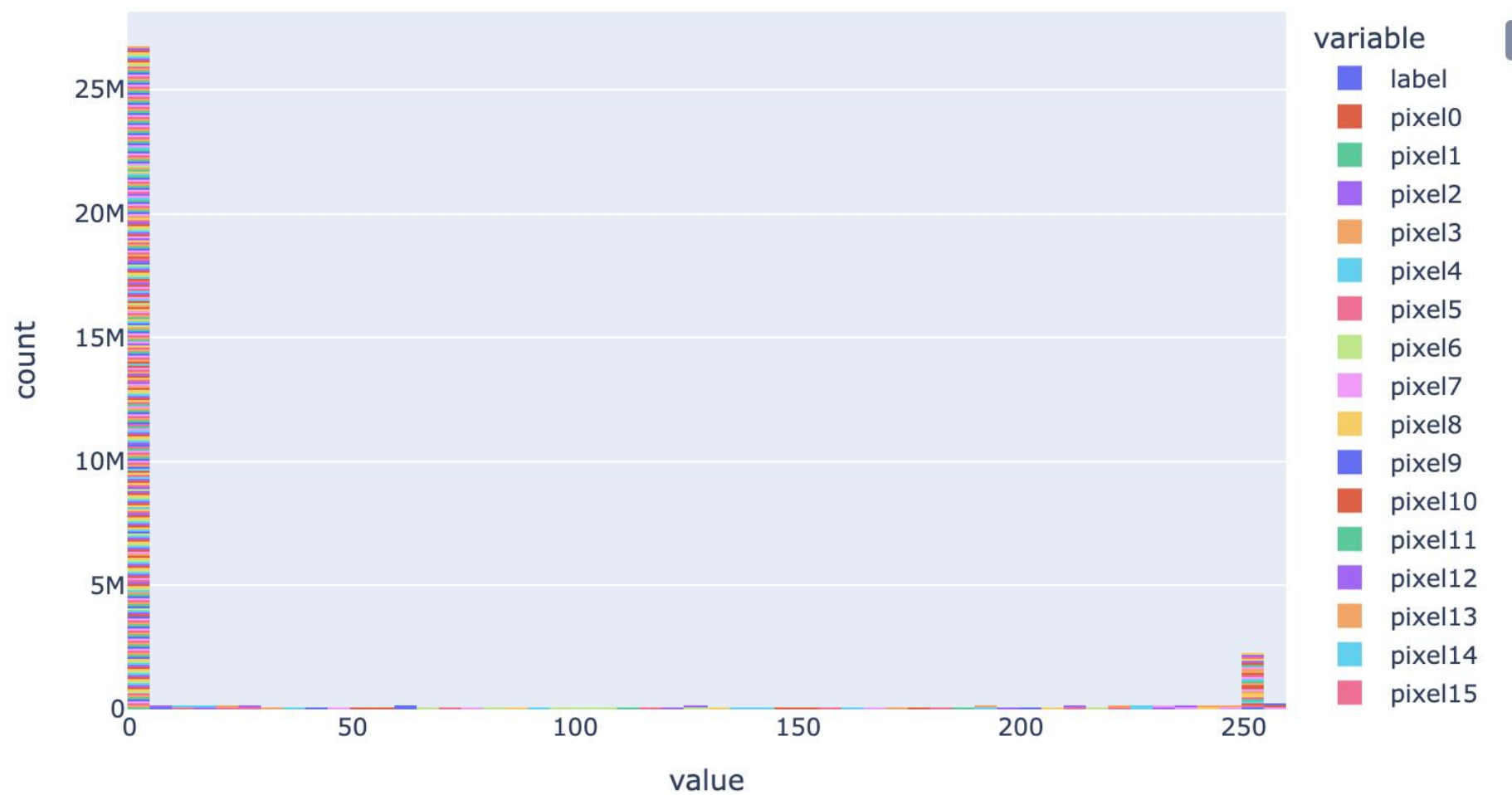
(рис.5)

Logarithmic histogram



(рис.6)

Interactive Histogram



(рис.7)

- Проверка:

а. пропусков (рис. 8)

```
nulls = train_data.isnull().sum()
```

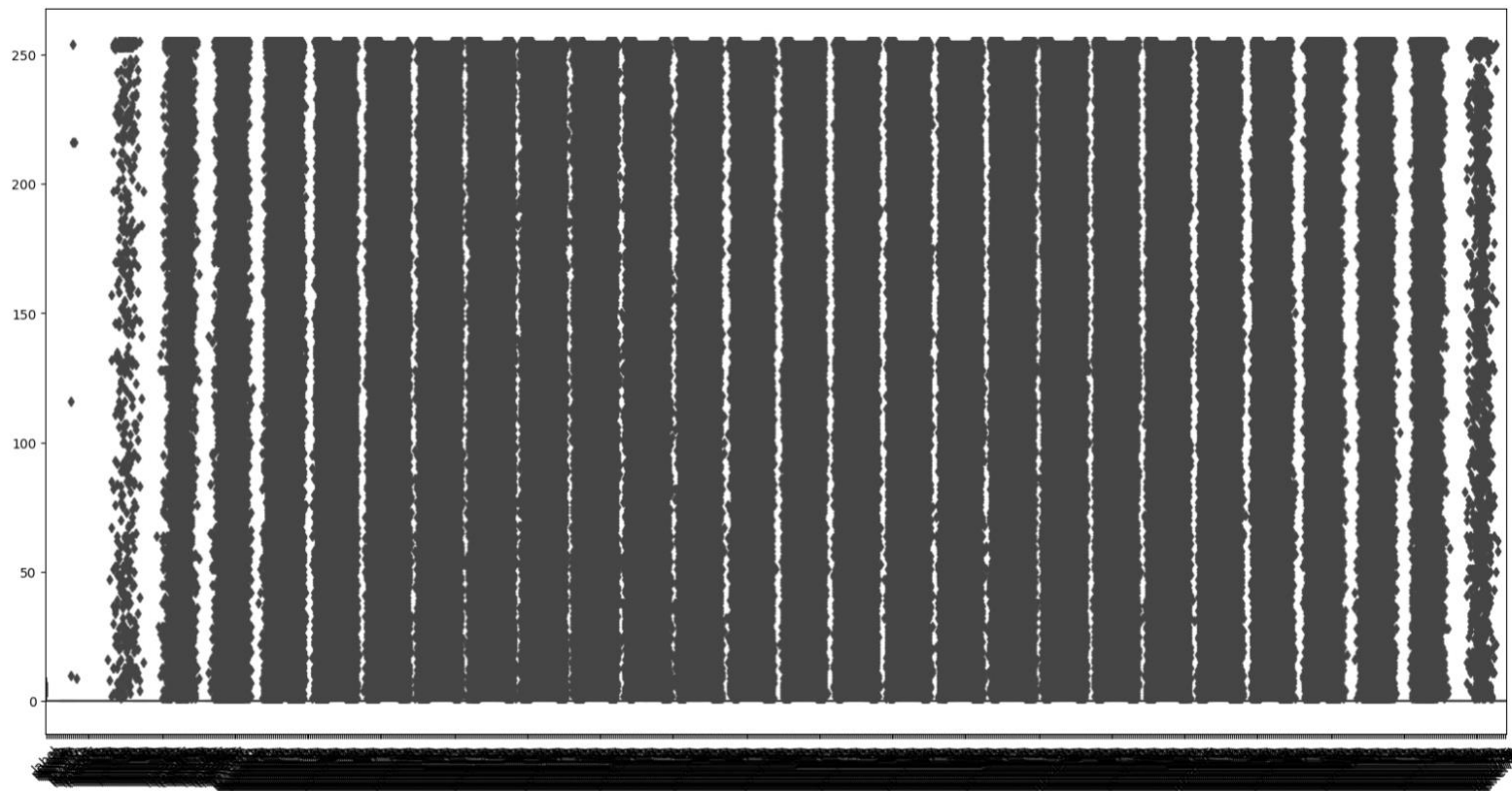
```
nulls = nulls>nulls>0]
```

```
print(nulls)
```

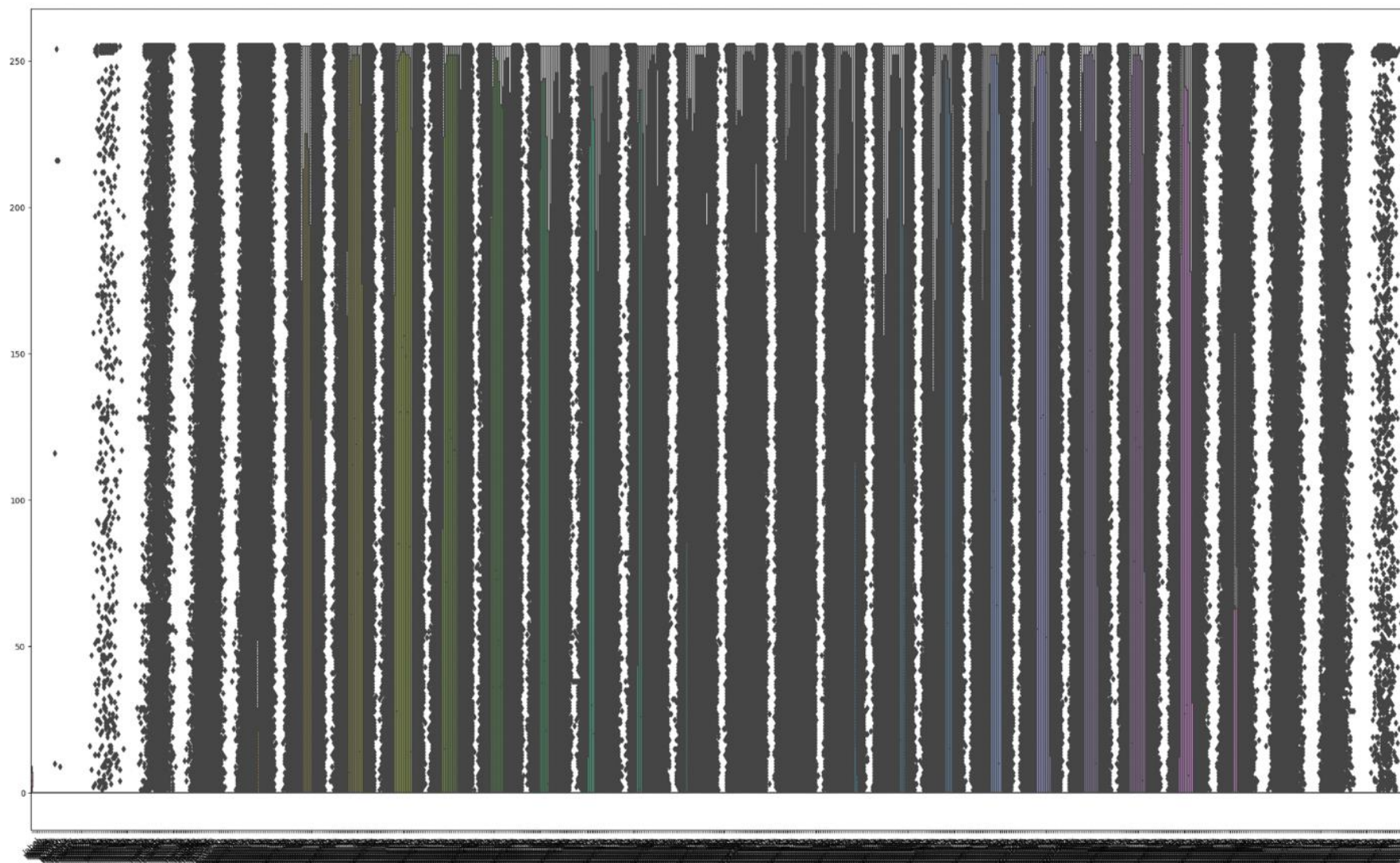
Series([], dtype: int64)

(рис. 8)

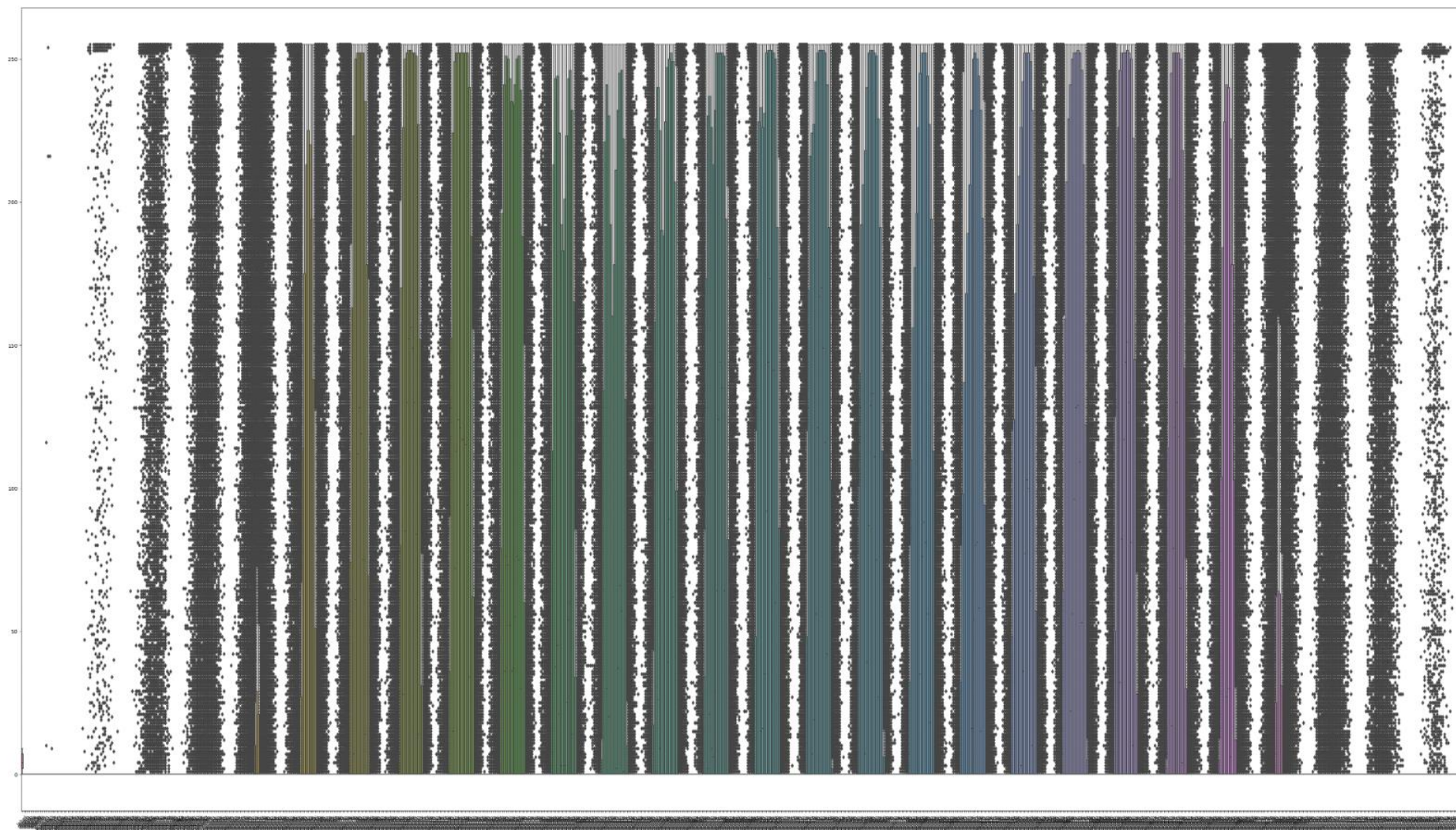
б. выбросов (рис.9-11):



(рис.9)



(рис.10)



(рис.11)

2. Выбор модели:

- Описание используемой архитектуры нейронной сети (например, MLP, CNN).

Архитектура сети представляет собой MLP со следующей структурой:

1. Входной слой

Преобразование данных:

- Изображения 28x28 пикселей преобразуются в вектор размерности 784 (28×28)
- Делается автоматически в forward методом `x.view(-1, 28*28)`

2. Основные слои архитектуры:

Слой	Тип	Размерность	Активация	Доп. обработка
fc1	Полносвязный	784 → 512	ReLU	BatchNorm + Dropout(0.5)
fc2	Полносвязный	512 → 128	ReLU	BatchNorm + Dropout(0.5)
fc3	Полносвязный	128 → 64	ReLU	BatchNorm + Dropout(0.5)
fc4	Полносвязный	64 → 10	Нет	-

(ReLU применяется неявно между слоями (в forward методе);

на выходе используются raw logits (без Softmax), так как CrossEntropyLoss включает Softmax внутри себя)

3. Регуляризация и нормализация:

Batch Normalization:

- bn1, bn2, bn3 – нормализуют выходы после каждого полносвязного слоя.
- Ускоряют обучение и улучшают стабильность.

Dropout:

- Вероятность 0.5 после каждого слоя.
- Предотвращает переобучение.

4. Выходной слой

- 10 нейронов (по числу классов цифр 0-9)
- Softmax применяется автоматически внутри CrossEntropyLoss

5. Forward

- Распрямление: изображение [1, 28, 28] → вектор [784].
- Полносвязные слои (fc1, fc2, fc3):
Линейные преобразования ($W \cdot x + b$) с уменьшением размерности: 784 → 512 → 128 → 64.

После каждого: BatchNorm (нормализация), ReLU (активация), Dropout (регуляризация).

- Выходной слой (fc4): $64 \rightarrow 10$ (логиты для 10 классов).

6. Backward:

- Реализован в модуле torch

- Обоснование выбора активационных функций, оптимизатора и loss-функции.

ReLU:

- Вычислительная эффективность: выполняется быстрее, чем сигмоида или гиперболический тангенс (не требует вычисления экспонент).
- Устранение проблемы затухающих градиентов: не "сжимает" градиенты при больших значениях, что ускоряет обучение (в отличие от сигмоиды)
- Разреженность активаций: обнуляет отрицательные значения, что делает сеть более разреженной и устойчивой к шуму.
- Лучшая сходимость для задач классификации изображений.

SGD:

- Простота и надежность: хорошо работает на задачах с четкими градиентами (как MNIST).
- Контролируемое обучение: позволяет точно настраивать learning rate и momentum.
- Меньше гиперпараметров: По сравнению с Adam или RMSprop, SGD требует меньше настроек.
- Хорошо работает с CrossEntropy, в сочетании с ReLU дает стабильное обучение.

CrossEntropy:

- Оптимальна для классификации: лучше штрафует за ошибки, чем MSE.
- Чувствительность к вероятностям: сильнее "наказывает" за уверенные, но неправильные предсказания.

- Гиперпараметры (число слоев, размер батча, эпох)

число слоев – 4 полносвязных слоя

размер батча – 32

кол-во эпох – 8

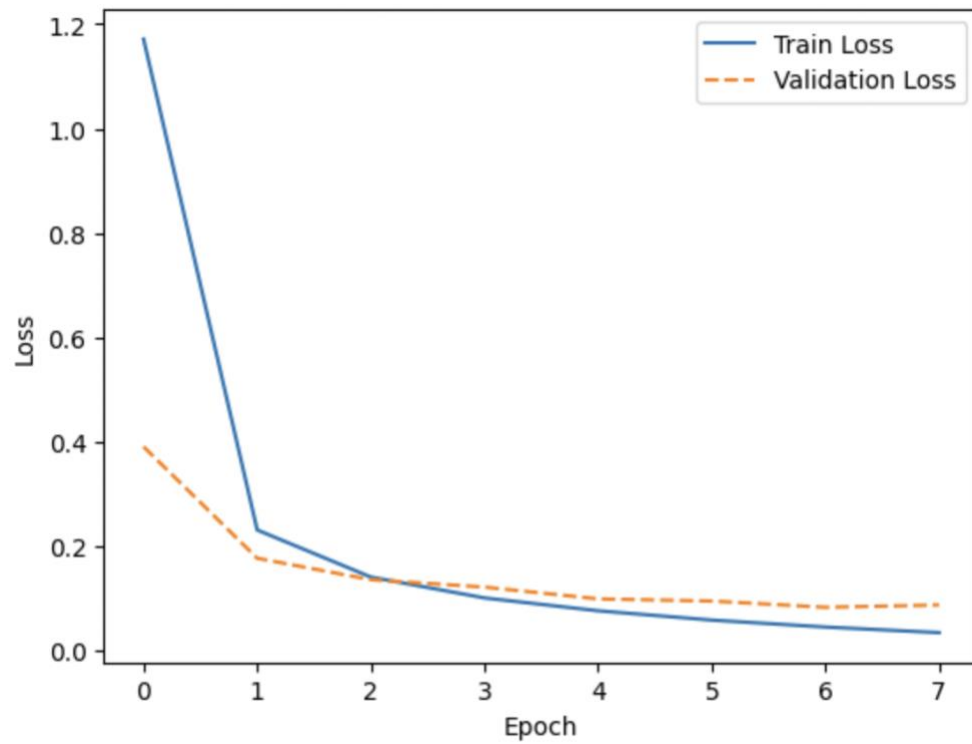
3. Результаты:

- Таблица с метриками (accuracy, loss, RMSE и др.)

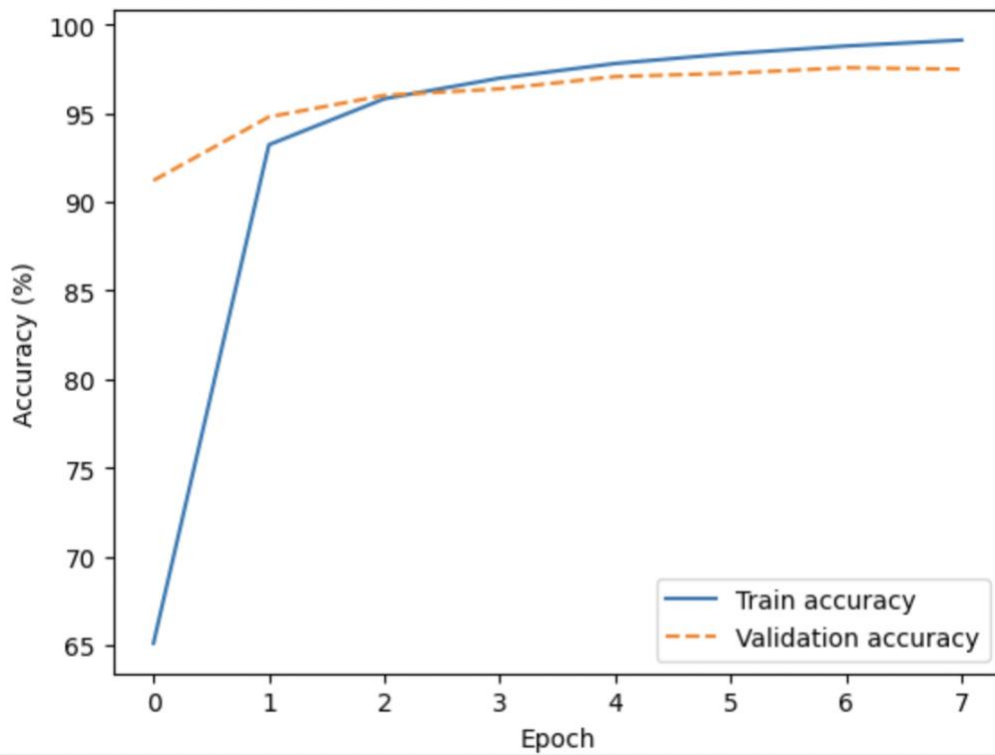
Epoch	Train Loss	Train accuracy	Valuate Loss	Valuate accuracy
0	1.1823	64.59%	0.4206	90.55%
1	0.2368	93.03%	0.1836	94.48%
2	0.1435	95.76%	0.1350	95.70%

3	0.1030	96.93%	0.1120	96.54%
4	0.0767	97.84%	0.0993	96.90%
5	0.0605	98.33%	0.0883	97.13%
6	0.0469	98.65%	0.0835	97.32%
7	0.0365	99.03%	0.0824	97.40%

- График обучения (loss и ассурасу на train/val) на отдельном рисунке (рис.12-13)



(рис.12)



(рис.13)

- Ссылка на submission в leaderboard – <https://www.kaggle.com/competitions/digit-recognizer/submissions>
аккаунт – AnastasiiaKotovaInt