

# Классификация объектов с помощью однопиксельной визуализации и нейронных сетей

А.А. Реутов<sup>1</sup>, Д.В. Бабухин<sup>1</sup>, Д.В. Сыч<sup>1</sup>

<sup>1</sup> Физический институт им. П. Н. Лебедева Российской академии наук, 119991, ГСП-1, Москва, Ленинский проспект, д.53

## Аннотация

Однопиксельная визуализация — это перспективный метод получения изображений без использования многопиксельных матриц. В отличие от стандартной фотографии, искомое изображение здесь не регистрируется напрямую матричным сенсором, а вычисляется. В последнее время для решения этой задачи начали применяться методы машинного обучения. В данной работе мы показываем возможности применения сверточных нейронных сетей в однопиксельной визуализации для классификации объектов по существенно неполному набору измерений. Мы находим зависимость точности классификации от различных параметров сэмпирования объекта. Предложенные методы могут использоваться в реальных устройствах в качестве эффективного программного обеспечения.

**Ключевые слова:** однопиксельная визуализация, классификация объектов без восстановления изображений, сверточные сети.

**Цитирование:** Реутов, А.А. Классификация объектов с помощью однопиксельной визуализации и нейронных сетей / А.А. Реутов, Д.В. Бабухин, Д.В. Сыч // Компьютерная оптика. – 2025. – Т. 49, № 3. – С. 517-524. – DOI: 10.18287/2412-6179-CO-1538.

**Citation:** Reutov AA, Babukhin DV, Sych DV. Object classification using a single-pixel camera and neural networks. Computer Optics 2025; 49(3): 517-524. DOI: 10.18287/2412-6179-CO-1538.

## Введение

Традиционные методы формирования изображений используют матричные детекторы, обеспечивающие пространственное разрешение целевого объекта. Однако использование матричных детекторов вне спектрального диапазона чувствительности кремния сталкивается с серьезными технологическими трудностями. Альтернативой матричным детекторам является использование детектора без пространственного разрешения в сочетании с пространственной модуляцией освещения объекта и вычислительными алгоритмами восстановления изображения – метод, получивший название однопиксельная визуализация [1, 2]. Одно из важных преимуществ однопиксельной визуализации состоит в возможности гораздо более широкого выбора одного детектора по сравнению с матрицей детекторов. Например, так можно получать изображения с помощью однофотонного детектора в ближней инфракрасной области [3] или использовать фоторезистор с широким спектральным диапазоном [4].

Процесс сбора данных в однопиксельной визуализации состоит в освещении объекта последовательностью пространственно-неоднородных световых паттернов и измерения соответствующего им интегрального количества света, отраженного от объекта, характеризующего соответствие паттерна объекту (сэмплирование объекта последовательностью паттернов). Вычислительная задача однопиксельной визуализации заключается в восстановлении изображения по известному набору паттернов и измеренным данным. В последние годы методы машинного обучения

продемонстрировали большой потенциал в решении этой задачи [5–10].

В практических применениях задача восстановления изображения, то есть получения пространственного распределения интенсивности света от объекта, может быть лишь промежуточным шагом. Типичным примером является классификация объектов [8]. В стандартном подходе эта задача решается в два шага: на первом шаге мы получаем изображение объекта, а на втором шаге вычисляется функция от этого изображения, по результату которой объект относится к одной из нескольких групп. Применение однопиксельной визуализации может решить задачу классификации объектов без промежуточного шага в виде восстановления изображения.

В данной работе мы демонстрируем классификацию изображений методом однопиксельной визуализации с помощью сверточных нейронных сетей (СНС). Наша основная идея применимости СНС в однопиксельной визуализации в данном случае состоит в том, что на первом шаге стандартного подхода к классификации объектов мы получаем сильно избыточную информацию об объекте (изображение объекта), которую далее сводим к гораздо меньшей (классу объекта). Поэтому мы можем изначально сэмплировать объект не с целью промежуточного получения изображения, а напрямую классифицировать набор сэмплов. Из такой постановки задачи появляется возможность сократить количество сэмплов и уменьшить время экспонирования объекта. Мы анализируем возможности такого ускорения, и находим оптимальные параметры всей схемы однопиксельной

визуализации для решения задачи классификации объектов на примере набора рукописных цифр MNIST.

Статья организована следующим образом: в параграфе 1 приводятся основные сведения о методе однопиксельной визуализации, в параграфе 2 обсуждается архитектура сверточной нейронной сети в контексте применения для однопиксельной классификации, в параграфе 3 показаны результаты классификации изображений, в параграфе 4 анализируется влияние шума детектора на точность классификации.

### 1. Однопиксельная визуализация

Устройство для однопиксельной визуализации (однопиксельная камера) включает в себя два основных компонента. Первый компонент – это пространственный модулятор света, позволяющий создать неоднородное структурированное освещение объекта (спроецировать на объект серию световых паттернов), или же структурировать сфокусированное на пространственном модуляторе света изображение объекта (пропустить изображение объекта через серию паттернов). Пространственный модулятор света может быть реализован по-разному, например, это может быть цифровой микрзеркальный модулятор, состоящий из массива маленьких управляемых зеркал, каждое из которых может принимать два положения с разными углами отклонения («включено» и «выключено»). Это позволяет произвольно маскировать свет, освещающий объект или отражённый от объекта.

Далее структурированный свет от объекта посредством собирающей линзы фокусируется на втором компоненте однопиксельной камеры – однопиксельном детекторе света (например, фотодиоде), который позволяет измерить общую интенсивность света для каждого светового паттерна. Визуализируемый объект последовательно освещается с использованием  $M$  различных световых паттернов, что позволяет измерить  $M$  значений интенсивности света. Далее для получения изображения требуется обработать набор из паттернов и соответствующих им интенсивностей.

Рассмотрим математическую модель процесса однопиксельной визуализации. Пусть изображение визуализируемого объекта имеет  $W$  (width) и  $H$  (height) пикселей по ширине и высоте соответственно, но далее мы будем использовать только общее число пикселей  $N = W \times H$  (такой же будет и размерность световых паттернов). Тогда, если в процессе работы камеры мы использовали  $M$  паттернов, на выходе мы имеем вектор  $S = (s_1, \dots, s_M)$  интенсивностей для каждого паттерна. Если представить набор из  $M$  матриц световых паттернов в векторизованном виде  $P = (\vec{P}^1, \dots, \vec{P}^M)^T$ , где  $\vec{P}^i = (p_1^i, \dots, p_{WH}^i)$  – векторизованная матрица  $i$ -го паттерна, то для векторизованного изображения объекта  $I = (i_1, \dots, i_{WH})$  взаимосвязь полученных отсчётов интенсивностей и пикселей изображения имеет вид

$$S = PI. \quad (1)$$

Если бы мы в результате работы камеры получили  $M=N$  значений интенсивностей без каких-либо отклонений от такого модельного описания (без шума аппаратуры и с неограниченным временем набора статистики фотоотсчётов), то искомое изображение можно было бы получить, выполнив умножение на обратную матрицу паттернов  $I = P^{-1}S$ . Далее это изображение можно классифицировать стандартными методами классификации, например, с использованием СНС. Однако на практике такой подход не оптимален по нескольким причинам. Во-первых, измеренные величины интенсивности содержат шум, поэтому восстановленное таким образом изображение также содержит шум, что ограничивает точность классификации даже для полного набора измерений. Во-вторых, любой алгоритм восстановления сам по себе также допускает некоторую неточность и вносит искажения в изображение, особенно сильные при малом количестве сэмплов, что затрудняет дальнейшую классификацию. Поэтому возможность избежать шага, связанного с восстановлением изображения, может существенно упростить задачу классификации без необходимости совместной оптимизации промежуточного восстановления искаженного изображения и последующей классификации.

Как мы покажем ниже, задачу классификации можно эффективно решить при  $M \ll N$ , то есть при существенно неполном наборе измерений, за счет использования СНС для обработки результатов измерений интенсивности света. В нашей модели СНС-классификатора мы используем промежуточное представление размером 28 на 28, которое не соответствует изначальному изображению (т.е. отсутствует восстановление изображения), но удобно для обучения модели классификации.

### 2. Сверточные нейронные сети для однопиксельной классификации

Для классификации объектов по сигналам с однопиксельной камеры в начале были выбраны изображения и паттерны размером  $N = 28 \times 28 = 784$ . Изображения были взяты из стандартного набора данных MNIST – набор данных образцов рукописного написания цифр.

Мы разработали и обучили сверточную нейронную сеть (СНС) на входных данных  $\vec{S}$  и выходных данных разметки набора MNIST. Структурно СНС состоит из нескольких блоков, схематично представленных на рис. 1: входного полносвязного (decode\_lin) (блоки fc1, fc2), обратных сверточных блоков (decode\_conv) (блоки (deconv1), (deconv2), (deconv3), (deconv4)), сверточных блоков (conv1), (conv2), (conv3) и выходного полносвязного (fc). Сначала полносвязные слои преобразуют данные сигналов размера  $M$  в данные размера 288. Далее сверточные блоки сворачивают данные до размерности 128.

Финальный полносвязный блок преобразует данные с последнего сверточного слоя в выходные данные размера 10. По номеру элемента выходных данных, имеющего максимальное значение, определяется

цифра на изображении набора MNIST, с которого получены сигналы  $\tilde{S}$ , поданные на вход СНС. В Приложении приведён подробный список слоев СНС и использованных функций активации.

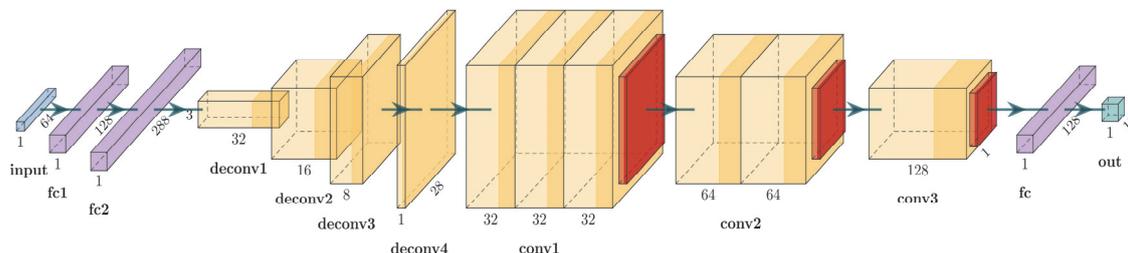


Рис. 1. Схематическое изображение архитектуры нейронной сети, использованной нами для классификации изображений

В начале для обучения были взяты наборы паттернов с числом паттернов 16, 32, 64 (соответственно, матрицы паттернов  $\hat{P}$  имели размер  $784 \times 16$ ,  $784 \times 32$  и  $784 \times 64$ ). Матрицы паттернов задавались как случайные бинарные матрицы со значениями 0 и 1, причем вероятность 0 варьировалась от 0,5 до 0,99. Отметим несколько особенностей использования паттернов со случайной структурой.

Во-первых, такие паттерны не обусловлены структурой и особенностями датасета (или класса примерных изображений, на которых далее будет работать однопиксельная камера), а также напрямую не связаны с алгоритмом обработки сигнала с однопиксельной камеры. В частности, вместо классификации рукописных цифр из набора MNIST могут быть использованы и другие алгоритмы, без внесения изменения в набор паттернов и экспериментальную реализацию однопиксельной камеры. Например, может быть выполнено восстановление изображения методами сжатого сэмпирования [1], или же проведена классификация изображения не по десяти классам, соответствующим рукописным цифрам, а по каким-либо другим особенностям изображения. То есть на одном и том же наборе сигналов с однопиксельного детектора могут быть выполнены разные алгоритмы его обработки, что делает такой подход достаточно универсальным. В других подходах, например в работе [8], используемый набор паттернов связан как с конкретным датасетом (MNIST), так и с конкретным алгоритмом обработки сигналов (классификатором). С одной стороны, это несколько улучшает точность классификации (96,1 % вместо 91,5 %). С другой стороны, это ограничивает однопиксельную камеру, запрограммированную на данный набор паттернов, одной конкретной задачей.

Во-вторых, случайные паттерны, в отличие от конкретно заданных паттернов, могут быть реализованы более широким классом экспериментальных методов. Для создания произвольных паттернов в большинстве экспериментальных работ по однопиксельной визуализации используются микрозеркальные пространственные модуляторы света. Это очень удобный инструмент, позволяющий создавать пат-

терны с высоким разрешением вплоть до 4К и менять их на частоте порядка 10 кГц. Тем не менее, полноценное управление таким устройством требует низкоуровневого программирования ПЛИС и технически трудоемко. Если же не требовать создания каких-либо конкретно заданных паттернов, то их можно реализовать другими методами. Например, можно использовать случайную спекл-структуру рассеянного когерентного света. Поэтому использование случайных паттернов может как упростить экспериментальную реализацию, так и повысить частоту смены паттернов.

Для всех запусков обучения было выбрано 30 эпох обучения с оптимизатором ADAM [11]. В оптимизаторе был выбран параметр скорость обучения  $lr=0,0003$  первые 15 эпох и  $lr=0,0001$  последние 15 эпох. В качестве валидационного набора были случайно отобраны 12000 изображений обучающего набора MNIST, обучение проводилось на оставшихся 48000 изображений. 10000 тестовых изображений использовались из тестового набора MNIST. В качестве функции потерь, определяющей расходимость между ожидаемым и предсказанным значением, была выбрана отрицательная логарифмическая функция правдоподобия. Обучение проводилось на i5-8250U CPU.

### 3. Результаты работы сверточной нейронной сети

Результаты классификации объектов методом однопиксельной визуализации с помощью обученных СНС представлены на рис. 2. Наилучшая точность распознавания рукописных цифр 94,1 % была получена в конфигурации с 64 паттернами. Отметим, что 64 паттерна составляют менее 9 % от полного набора паттернов для изображений 28 на 28. Также отметим, что даже использование всего 16 паттернов (чуть более 2 % от полного набора) показывает хороший результат (точность 78,7 %).

Далее рассмотрим вопрос об оптимизации соотношения нулей и единиц в используемых паттернах. При классификации изображений у нас может быть использована некоторая априорная информация о наборе возможных изображений. В частности, для

набора MNIST распределение светлых и темных пикселей не равномерно по всему изображению. Кроме того, сэмплирование изображений случайными паттернами с разными вероятностями нулей и единиц с последующим восстановлением изображения методами сжатого сэмплирования показывает, что оптимальный набор паттернов получается при очень небольшой вероятности единиц [12, 13]. В частности,

для набора MNIST распределение светлых и темных пикселей не равномерно по всему изображению. Поэтому нами была обучена СНС на матрицах паттернов с неравновзвешенными значениями 1 и 0. Для этого были случайно сгенерированы матрицы с 10% и 1% единиц (соответственно 90% и 99% нулей). Оптимизатор, количество эпох, скорость обучения, функция потерь, набор данных не менялись.

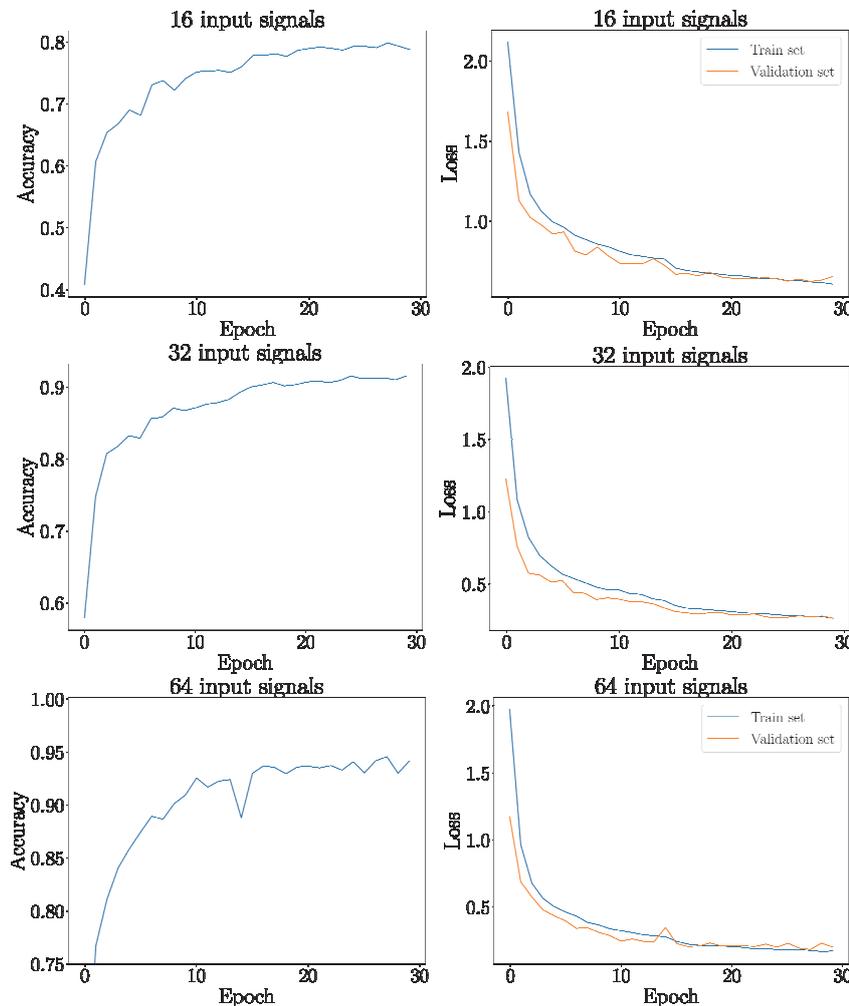


Рис. 2. Точность (Accuracy) обучения по тестовому набору и динамика функции потерь (Loss) в ходе обучения. Количество паттернов и входных сигналов на СНС – 16 (первая строка), 32 (вторая строка) и 64 (третья строка). Точность классификации, достигнутая к концу обучения, – 78,7%, 91,5% и 94,1% соответственно

Был выбран набор из 64 паттернов, т.к. при таком количестве была достигнута наилучшая точность распознавания символов в предыдущем пункте. Результаты классификации представлены на рис. 3.

Характерно, что СНС за 30 эпох существенно переобучается (функция потерь на тестовом наборе дает значения существенно выше, чем на тренировочном наборе), к началу переобучения (8–9 эпоха). В обоих случаях достигается точность около 93%, что слабо отличается от результатов для равновзвешенной матрицы 64 паттернов.

Для обученной модели можно было бы ожидать, что в промежуточных слоях, а именно после блока

(decode\_conv) в промежуточных значениях размера 28 на 28, могло быть достигнуто полное восстановление изображения (что могло бы быть интересным результатом, позволяющим не находить под веса обученной нейросети специально подобранные паттерны, а ограничиться случайными). Тем не менее восстановления изображения в промежуточных слоях не достигается, как показано на рис. 4.

#### 4. Шум детектора в однопиксельной классификации

Как мы отмечали ранее, в реальном эксперименте сигнал однопиксельного детектора всегда содержит шумовую составляющую. Для апробации результатов

нашей работы на практике целесообразно проанализировать влияние шума на точность классификации.

Для этого рассмотрим шум как случайные флуктуации сигнала в виде:

$$S' = P \cdot I * N[\mu, \sigma], \tag{2}$$

где \* – поточечное умножение, а  $N[\mu, \sigma]$  – матрица случайных величин размера  $M \times N$ , где каждый эле-

мент – случайная величина с гауссовым распределением со средним  $\mu = 1$  и среднеквадратичным  $\sigma = 0,01, 0,03, 0,06$  и  $0,1$ .

На таких зашумленных сигналах с выбранными 64 паттернами были получены обученные веса СНС. Оптимизатор, количество эпох, скорость обучения, функция потерь, набор данных были оставлены теми же, что и в предыдущем процессе обучения.

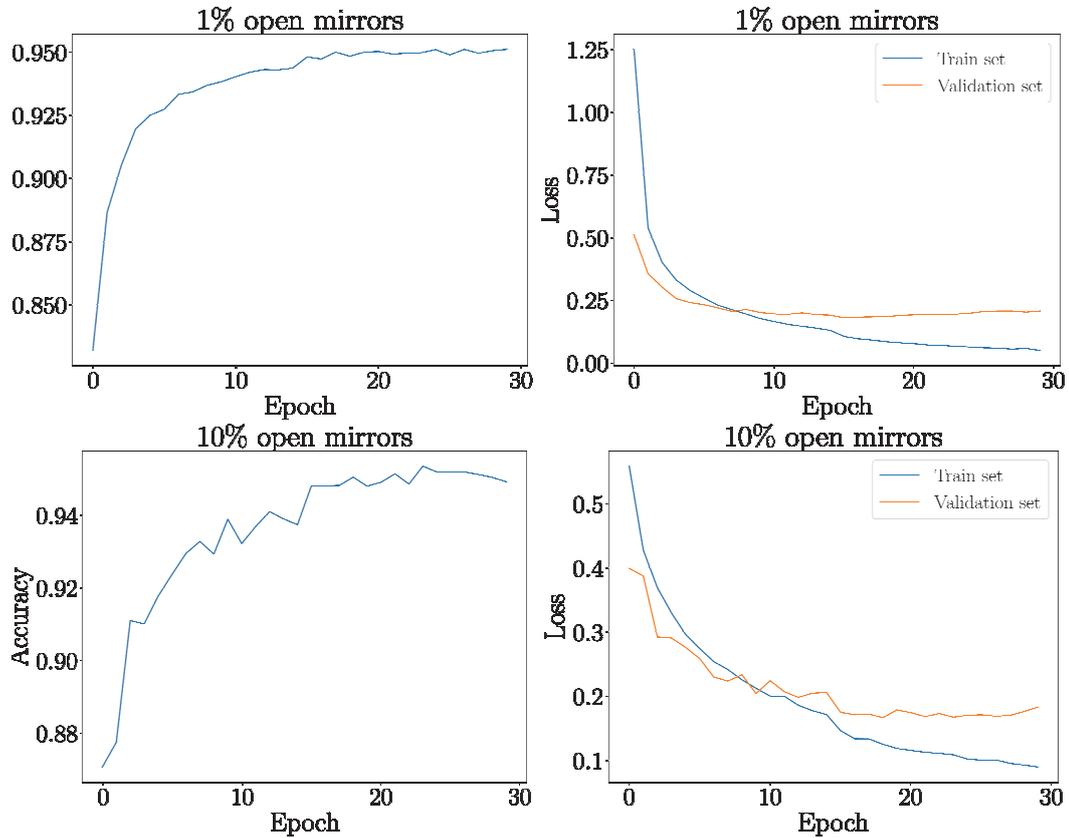


Рис. 3. Точность (Accuracy) обучения по тестовому набору и динамика функции потерь (Loss) в ходе обучения. Количество паттернов и входных сигналов на СНС – 64. Соотношение 1 и 0 в матрице паттернов – 1/99 (верхний ряд) и 1/9 (нижний ряд). Точность к моменту переобучения, – 93,1% и 93,2% соответственно

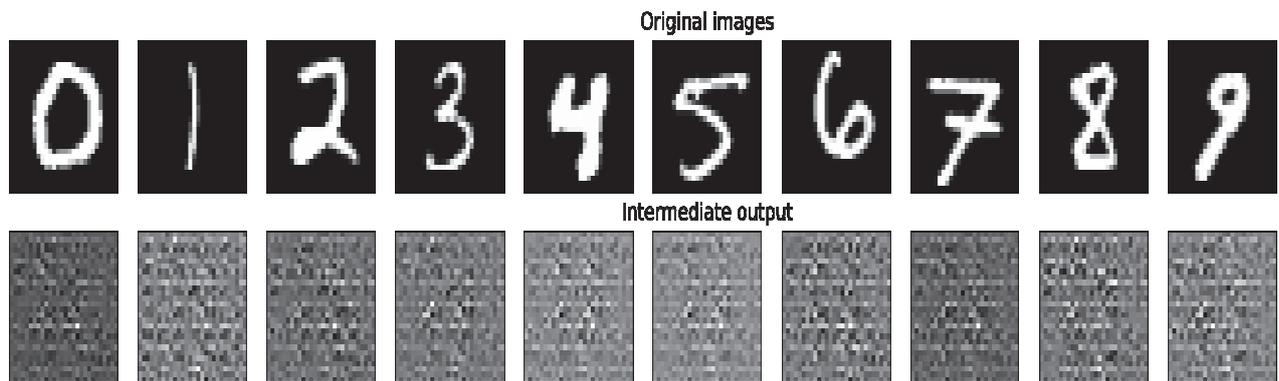


Рис. 4. Сравнение исходных изображений и промежуточных изображений после блока (decode\_conv)

Результаты представлены на рис. 5. При небольшом шуме  $\sigma = 1\%$  результаты аналогичны тем, что получены в отсутствие шума – 94%. Приемлемые результаты наблюдаются при средней вели-

чине шума  $\sigma = 3\%$ , где точность распознавания символов достигает почти 91%. Большой шум  $\sigma = 10\%$  дает значительно худшие результаты, точность падает до менее 70%.

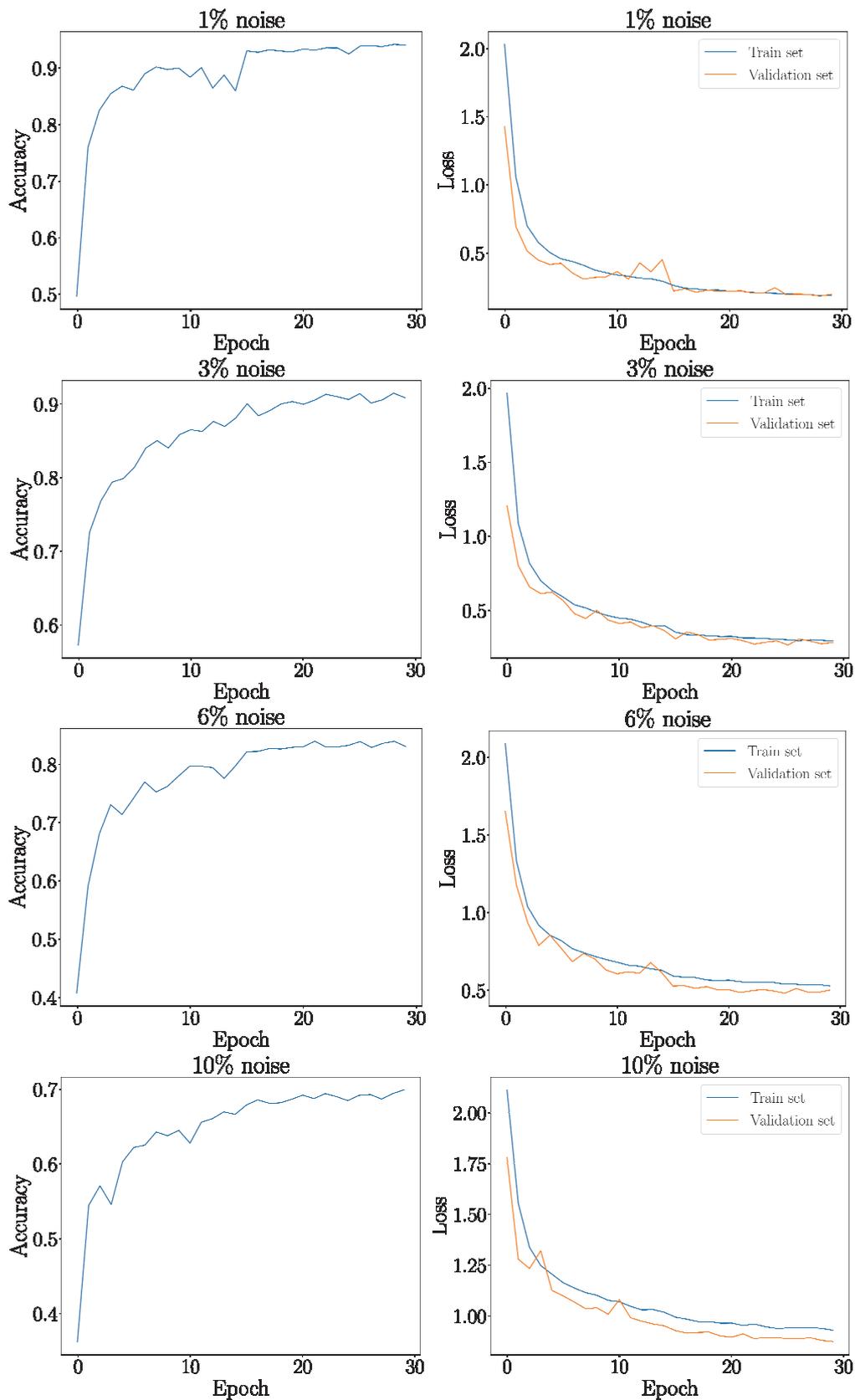


Рис. 5. Точность (Accuracy) обучения по тестовому набору и динамика функции потерь (Loss) ходе обучения. Количество паттернов и входных сигналов на ЧНС – 64. Уровень шума  $\sigma = 1\%$ ,  $3\%$ ,  $6\%$ ,  $10\%$  (первый, второй, третий и четвертый ряд). Точность к концу обучения –  $94\%$ ,  $90,8\%$ ,  $83\%$  и  $69,9\%$  соответственно

### Заключение

В данной работе мы продемонстрировали возможность распознавания и классификации объектов с помощью метода однопиксельной визуализации с применением сверточных нейронных сетей. На наборе изображений рукописных цифр MNIST мы показали, что можно использовать существенно неполные измерения объекта и при этом получать достаточно высокую точность классификации. Так, с использованием всего лишь 32 паттернов (примерно 4% от полного набора) можно достичь точность классификации выше 90%. Также мы продемонстрировали, что реалистичный шум детектора в виде случайных гауссовских флуктуаций в несколько процентов от величины сигнала несущественно влияет на точность классификации. Таким образом, применение нейронных сетей для задач, связанных с распознаванием объектов, в контексте однопиксельной визуализации является крайне перспективным направлением как с точки зрения существенного сокращения количества необходимых измерений и времени обработки сигнала, так и с точки зрения устойчивости к шумам детектирования. В качестве продолжения данного исследования стоит обратить внимание на сравнение данного метода с современными классификаторами [14], совмещенными с методами полного однопиксельного восстановления изображения, а также анализ для разных подходов точности классификации и особенностей практической реализации.

### Благодарности

Исследование выполнено за счет гранта Российского научного фонда № 23-22-00381, <https://rscf.ru/project/23-22-00381/>.

### References

[1] Duarte MF, Davenport MA, Takhar D, Laska JN, Sun T, Kelly KF, Baraniuk RG. Single-pixel imaging via compressive sampling. *IEEE Signal Process Mag* 2008; 25(2): 83-91. DOI: 10.1109/MSP.2007.914730.

- [2] Gibson GM, Johnson SD, Padgett MJ. Single-pixel imaging 12 years on: a review. *Opt Express* 2020; 28(19): 28190-28208. DOI: 10.1364/OE.403195.
- [3] Shcherbatenko M, Elezov M, Manova N, Sedykh K, Korneev A, Korneeva Yu, Dryazgov M, Simonov N, Feimov A, Goltsman G, Sych D. Single-pixel camera with a large-area microstrip superconducting single photon detector on a multimode fiber. *Appl Phys Lett* 2021; 118(18): 181103. DOI: 10.1063/5.0046049.
- [4] Aguilar RA, Hermosa N, Soriano MN. Low-cost Fourier ghost imaging using a light-dependent resistor. *Am J Phys* 2019; 87(12): 976-981. DOI: 10.1119/10.0000163.
- [5] Edgar MP, Gibson GM, Bowman RW, Sun B, Radwell N, Mitchell KJ, Welsh SS, Padgett MJ. Simultaneous real-time visible and infrared video with single-pixel detectors. *Sci Rep* 2015; 5(1): 10669. DOI: 10.1038/srep10669.
- [6] Higham CF, Murray-Smith R, Padgett MJ, Edgar MP. Deep learning for real-time single-pixel video. *Sci Rep* 2018; 8(1): 2369. DOI: 10.1038/s41598-018-20521-y.
- [7] Zhang Y, Edgar MP, Sun B, Radwell N, Gibson GM, Padgett MJ. 3D single-pixel video. *J Opt* 2016; 18(3): 035203. DOI: 10.1088/2040-8978/18/3/035203.
- [8] Zhang Z, Li X, Zheng S, Yao M, Zheng G, Zhong J. Image-free classification of fast-moving objects using “learned” structured illumination and single-pixel detection. *Opt Express* 2020; 28(9): 13269-13278. DOI: 10.1364/OE.392370.
- [9] Mur AL, Leclerc P, Peyrin F, Ducros N. Single-pixel image reconstruction from experimental data using neural networks. *Opt Express* 2021; 29(11): 17097-17110. DOI: 10.1364/OE.424228.
- [10] Xie S, Peng L, Bian L. Large-scale single-pixel imaging via deep learning. *Proc SPIE* 2023; 12317: 1231703. DOI: 10.1117/12.2643014.
- [11] Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv Preprint*. 2017. Source: <<https://arxiv.org/abs/1412.6980>>. DOI: 10.48550/arXiv.1412.6980.
- [12] Sych DV. Optimization of compressed sampling in single-pixel imaging. *Bull Lebedev Phys Inst* 2024; 51: 202-205. DOI: 10.3103/S1068335624600463.
- [13] Sych D. Influence of detector noise on compressed sampling single-pixel imaging. *J Russ Laser Res* 2024; 45(3): 286-294. DOI: 10.1007/s10946-024-10213-6.
- [14] Baldominos A, Saez Y, Isasi P. A survey of handwritten character recognition with mnist and emnist. *Appl Sci* 2019; 9(15): 3169. DOI: 10.3390/app9153169.

### Приложение

Структура модели:

CNN( (decode\_lin): Sequential( (0): Linear(in\_features=64, out\_features=128, bias=True) (1): ReLU(inplace=True) (2): Linear(in\_features=128, out\_features=288, bias=True) (3): ReLU(inplace=True) ) (unflatten): Unflatten(dim=1, unflattened\_size=(32, 3, 3)) (decode\_conv): Sequential( (0): ConvTranspose2d(32, 16, kernel\_size=(3, 3), stride=(2, 2)) (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (2): ReLU(inplace=True) (3): ConvTranspose2d(16, 8, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), output\_padding=(1, 1)) (4): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (5): ReLU(inplace=True) (6): ConvTranspose2d(8, 1, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), output\_padding=(1, 1)) ) (conv1): Sequential( (0): Conv2d(1, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): ReLU() (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (3): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1)) (4): ReLU() (5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (6): Conv2d(32, 32, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1)) (7): ReLU() (8): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (9): MaxPool2d(kernel\_size=2, stride=2, padding=0, dila-

tion=1, ceil\_mode=False) (10): Dropout(p=0.25, inplace=False) ) (conv2): Sequential( (0): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): ReLU() (2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (3): Conv2d(64, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1)) (4): ReLU() (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (6): Conv2d(64, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1)) (7): ReLU() (8): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (9): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False) (10): Dropout(p=0.25, inplace=False) ) (conv3): Sequential( (0): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): ReLU() (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True) (3): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False) (4): Dropout(p=0.25, inplace=False) ) (fc): Sequential( (0): Linear(in\_features=128, out\_features=10, bias=True) ) )

### *Сведения об авторах*

**Реутов Алексей Алексеевич**, 1995 года рождения, окончил Московский физико-технический институт, младший научный сотрудник в Физическом институте им. П.Н. Лебедева РАН. Область научных интересов: квантовая связь, однопиксельная визуализация, машинное обучение в задачах физики. E-mail: [aleksey.reutov@phystech.edu](mailto:aleksey.reutov@phystech.edu)

**Бабухин Данила Валерьевич**, 1995 года рождения, окончил Московский государственный университет имени М.В. Ломоносова, младший научный сотрудник в Физическом институте им. П.Н. Лебедева РАН. Область научных интересов: квантовая связь, однопиксельная визуализация, машинное обучение в задачах физики. E-mail: [dv.babukhin@gmail.com](mailto:dv.babukhin@gmail.com)

**Сыч Денис Васильевич**, 1980 года рождения, окончил Московский государственный университет имени М.В. Ломоносова, кандидат физико-математических наук, старший научный сотрудник Физического института им. П.Н. Лебедева РАН. Область научных интересов: квантовая информация, квантовая и классическая оптика, однопиксельная визуализация. E-mail: [denis.sych@gmail.com](mailto:denis.sych@gmail.com)

ГРНТИ: 28.23.15

Поступила в редакцию 10 апреля 2024 г. Окончательный вариант – 18 октября 2024 г.

---

# Object classification using a single-pixel camera and neural networks

A.A. Reutov<sup>1</sup>, D.V. Babukhin<sup>1</sup>, D.V. Sych<sup>1</sup>

<sup>1</sup> P.N. Lebedev Physical Institute, RAS, 53 Leninskiy Prospekt, Moscow, 119991, Russia

## Abstract

Single pixel imaging is a promising method of imaging without using multi-pixel matrices. Unlike traditional methods, the image here is not directly registered, but computed. Recently, machine learning techniques have started to be used to solve this problem. In this paper, we show the potential application of convolutional neural networks in single-pixel imaging to classify objects from a substantially incomplete set of measurements. We find the dependence of classification accuracy on various object sampling parameters. The proposed methods can be used in real devices as efficient software.

**Keywords:** single-pixel imaging, object classification without image restoration, convolutional networks.

**Citation:** Reutov AA, Babukhin DV, Sych DV. Object classification using a single-pixel camera and neural networks. *Computer Optics* 2025; 49(3): 517-524. DOI: 10.18287/2412-6179-CO-1538.

**Acknowledgements:** This work was supported by Russian Science Foundation (Project No. 23-22-00381, <https://rscf.ru/project/23-22-00381/>).

---

## Author's information

**Aleksei Alekseyevich Reutov**, (b. 1995), studied at Moscow Institute of Physics and Technology, junior researcher at P.N. Lebedev Physical Institute of the Russian Academy of Sciences. Research interests: quantum communication, single-pixel imaging, machine learning in physics. E-mail: [aleksey.reutov@phystech.edu](mailto:aleksey.reutov@phystech.edu)

**Danila Valerevich Babukhin**, (b. 1995), studied at Lomonosov Moscow State University, junior researcher at P.N. Lebedev Physical Institute of the Russian Academy of Sciences. Research interests: quantum communication, single-pixel imaging, machine learning in physics. E-mail: [dv.babukhin@gmail.com](mailto:dv.babukhin@gmail.com)

**Denis Vasilevich Sych**, (b. 1980), studied at Lomonosov Moscow State University, PhD. in Physics and Mathematics, senior researcher at P.N. Lebedev Physical Institute of the Russian Academy of Sciences. Research interests: quantum information, quantum and classical optics, single-pixel imaging. E-mail: [denis.sych@gmail.com](mailto:denis.sych@gmail.com)

---

*Received April 10, 2024. The final version – October 18, 2024.*

---