# Скелетное представление толстых линий для классификации изображений

Григорьев А.Д., Коробов Н.С., Куцевол П.Н., Лукоянов А.С. и Жариков И.

Московский физико-технический институт

*Курс:* Машинное обучение
(практика, В. В. Стрижов)/весна 2019

# Цель исследования

## Задача

Для данной выборки растровых изображений рукописных цифр построить модель, оптимальным образом классифицирующую изображенный символ.

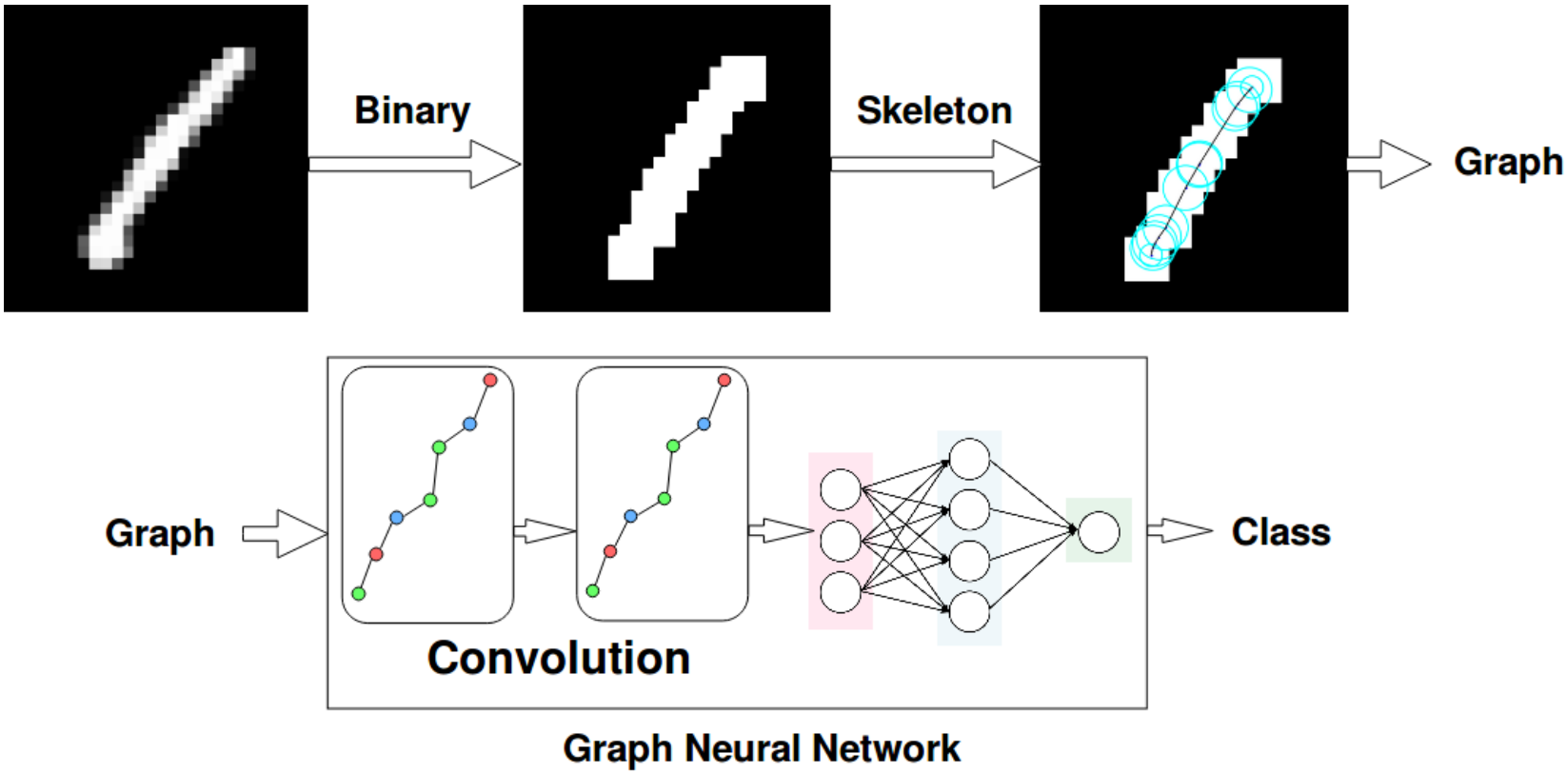
## Проблема

При высоком качестве классификации, существующие решения являются относительно ресурсоемкими. Долгое время обучения и предсказания, большие объемы занимаемой памяти критичны для мобильных устройств.

## Решение

Альтернативное представление растрового изображения - скелетное представление толстых линий. Такое представление снизит требования к ресурсам для обучения классификатора, а также повысит качество.

# Предлагается: скелеты, графы, классификация



# Существующие решения

**Решения сверточными нейронными сетями**

1. Yanai K., Tanno R., Okamoto K. Efficient mobile implementation of a cnn-based object recognition system //Proceedings of the 24th ACM international conference on Multimedia. – ACM, 2016. – С. 362-366.
2. Wan L. et al. Regularization of neural networks using dropconnect //International conference on machine learning. – 2013. – С. 1058-1066.

**Решения графовыми нейронными сетями**

1. Battaglia P. W. et al. Relational inductive biases, deep learning, and graph networks //arXiv preprint arXiv:1806.01261. – 2018.
2. Fey M. et al. SplineCNN: Fast geometric deep learning with continuous B-spline kernels //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – С. 869-877.

# Формальная постановка задачи классификации

Дана выборка из пар бинарного изображения $I \in \mathbb{R}^{n \times m \times 1}$ и класса изображения $y$:

$$D = (I_i, y_i)\ i = \{1, \ldots, n\}. \tag{1}$$

Бинарное изображение представляется в виде скелета с помощью функции:

$$s(I) : \mathbb{R}^{n \times m \times 1} \to \underbrace{\{\mathbb{R}^p, \mathbb{R}^p, \ldots, \mathbb{R}^p\}}_{I \text{ раз}}, \tag{2}$$

где $p$ - размерность пространства параметров скелета. Задана функция $g$ отображающая скелет $s(I)$ в граф $G(E, V)$, где каждой вершине $v_i$ соответсвует вектор признаков $h_i \in \mathbb{R}^k$ и вектор координат $x_i \in \mathbb{R}^d$, где $d$ - размерность пространства (2 для изображений), а $k$ - количество признаков.
Выборка $D$ изображений $\mathbf{I}$ с ответами $\mathbf{y}$ отображается в выборку графовых представлений:

$$D_G = (g(s(I_i)), y_i)\ i = \{1, \ldots, n\}. \tag{3}$$

# Оптимизация параметров

Модель классификации - суперпозиция функций $f \circ g \circ s$, где $f : G(E, V) \to \mathbb{R}^C$ – нейросеть, а $C$ – число классов. В качестве $f$ выберем графовые нейронные сети из множества: {MPNN, MoNet, $k$-GNN, SplineCNN}. Функции $g$ и $s$ зафиксируем. Задача имеет вид:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} L(D_G, \mathbf{w}|f), \tag{4}$$

где L - функция потерь Cross Entropy Loss

$$L(D_G, \mathbf{w}|f) = -\sum_{j=1}^{n} y_j \log \sigma(f(G_j(E, V)))_j \tag{5}$$

$$\sigma(z)_j = \frac{\exp z_j}{\sum_k^C \exp z_k}, \tag{6}$$

где $\sigma$ - Softmax.

# Message Passing Neural Network

Входные данные – графовые структуры, каждой из вершин сопоставляется вектор признаков $h$. Для каждой из вершин $T$ раз происходит обмен информации с ее соседями с помощью функции передачи сообщения $M$ с обновлением вектора признаков в каждой вершине с помощью функции $U$. Затем следует фаза вычитки информации из графа по всем вершинам – $R$.

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{7}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{8}$$

$$\hat{y} = R(h_v^T | v \in G) \tag{9}$$

# $k$-GNN

Пусть $(G, l)$ – граф с заданной раскраской.
Для каждого слоя $t \geq 0$ $k-$GNN вычисляется вектор признаков $f_k^{(t)}(s)$ $\forall s \in [V(G)]^k$, где $[V(G)]^k$ – множество всех подмножеств $V(G)$ мощности $k$. $\sigma$ – функция активации.

$$f_k^{(t)}(s) = \sigma(f_k^{(t-1)}(s) \cdot W_1^{(t)} + \sum_{w \in N(v)} f_k^{(t-1)}(s) \cdot W_2^{(t)}) \qquad (10)$$

Входные данные – граф $G(E, V)$, каждой из вершин $v_i$ которого сопоставлен вектор координат $u_i \in \mathbb{R}^d$ и вектор признаков $f_i \in \mathbb{R}^k$. Для каждой вершины по всем ее соседям вычисляется множество весов $\mathbf{w}$ в обобщенном пространстве координат:

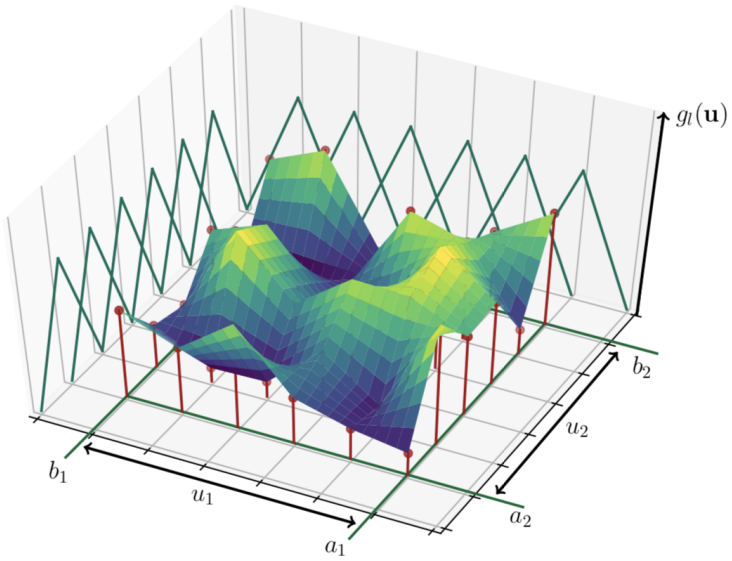
$$w_{\mu,\Sigma}(u) = \exp(-\frac{1}{2}(u-\mu)^T\Sigma^{-1}(u-\mu)), \tag{11}$$

где $\Sigma$ и $\mu$ - обучаемые параметры. Тогда операция свертки выглядит как:

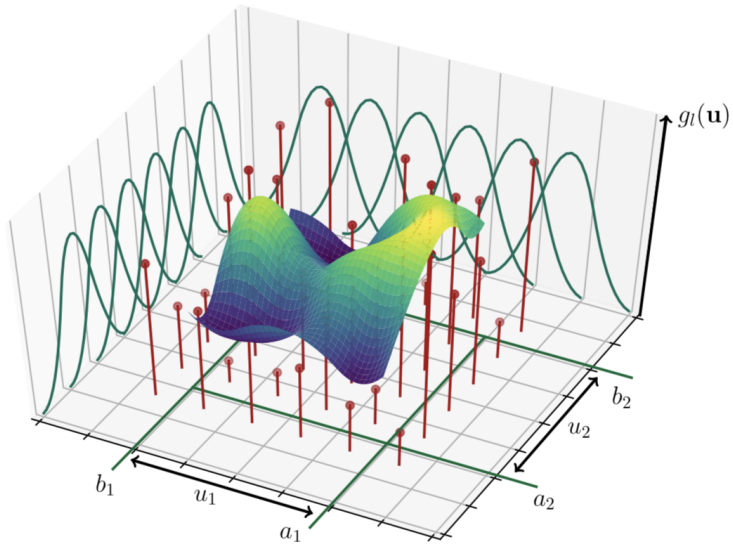
$$(f \star g) = \sum_{j=1}^{J}\sum_{k=1}^{n} g_j w_{\mu_k,\Sigma_k}(u(j,k))f_k \tag{12}$$

# SplineCNN

Модификация подхода MoNet. Функцией весов являются B-spline с обучаемыми коэффициентами с которыми суммируется заранее выбранная базисная функция.




(a) Linear B-spline basis functions




(b) Quadratic B-spline basis functions

Fey M. et al. SplineCNN: Fast geometric deep learning with continuous B-spline kernels, 2018

# Вычислительный эксперимент

## Цель эксперимента

Сравнить альтернативные модели классификации изображений с предложенным с точки зрения времени обучения до сходимости, времени вычисления класса, требуемой для обучения памяти и точности классификации.

## Базы данных

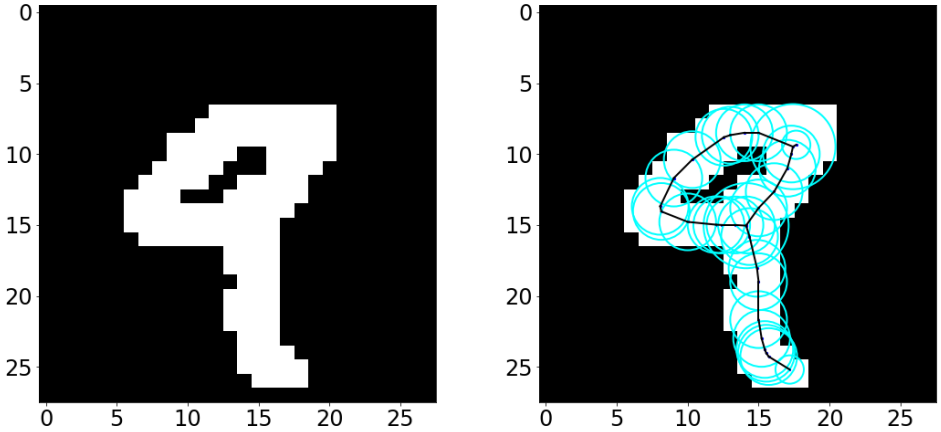
1. MNIST Skeleton – база данных скелетных представлений картинок MNIST.
2. MNIST Superpixels 75 – база данных графовых представлений над super pixel, полученных из базы данных MNIST.

# Метрики

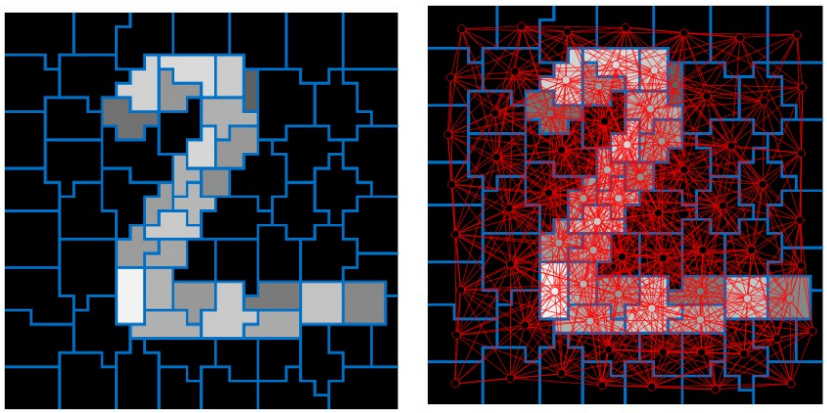
## Метрики эксперимента

Точность классификации – accuracy.
Время обучения до сходимости – время до остановки изменения функции потерь по некоторому порогу в секундах.
Время вычисления класса – время на предобработку изображения + время на работу классификатора в секундах .
Требуемая для обучения память – данные профилировщика о загруженности видео-памяти.

# База данных: примеры изображений рукописных цифр

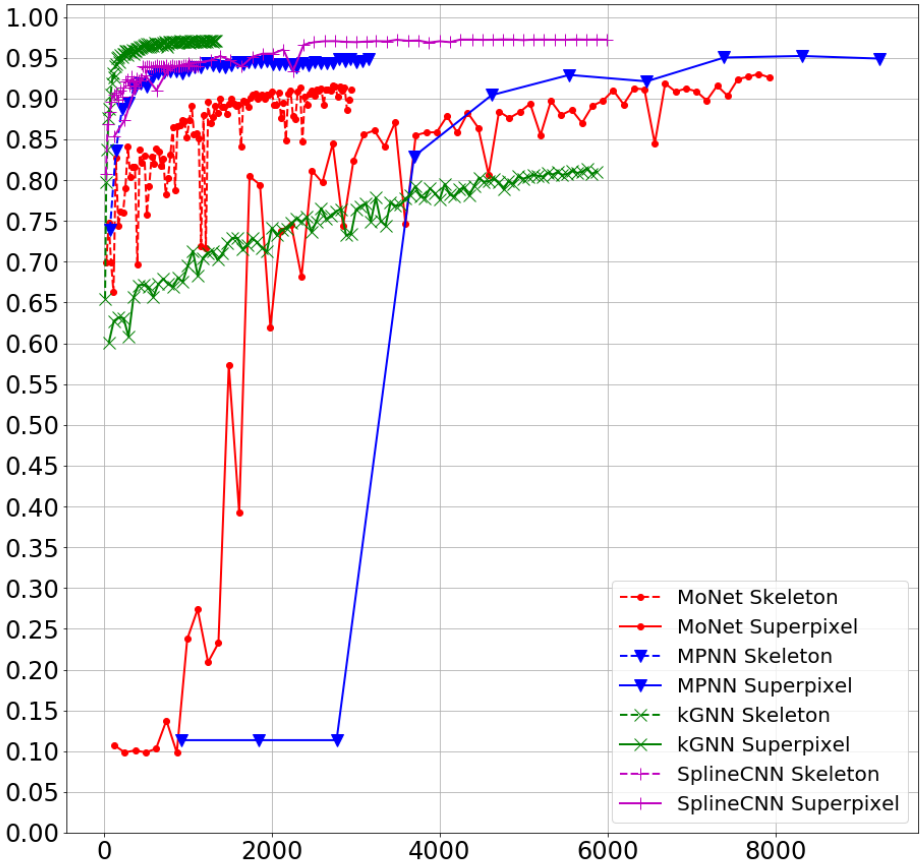



Скелетное представление цифры 9



Super pixels 75 и графовое представление над ними

# Результаты эксперимента




Зависимость *accuracy* от времени обучения *в секундах*

# Результаты эксперимента

Точность классификации по метрике *accuracy*

| | **MPNN** | ***k*-GNN** | **MoNet** | **SplineCNN** |
|---|---|---|---|---|
| Skeleton | 0.9486 | **0.971** | 0.9154 | 0.9413 |
| Superpixel | **0.9521** | 0.8143 | **0.9302** | **0.9724** |

Использование GPU в процессе обучения *в Гб*

| | **MPNN** | ***k*-GNN** | **MoNet** | **SplineCNN** |
|---|---|---|---|---|
| Skeleton | **1.50** | **0.34** | **0.32** | **0.73** |
| Superpixel | 10.76 | 0.85 | 1.54 | 1.04 |

Время предсказания на тесте с batch size 1 *в секундах*

| | **MPNN** | ***k*-GNN** | **MoNet** | **SplineCNN** |
|---|---|---|---|---|
| Skeleton | **147.2** | 47.48 | **56.2** | 38.8 |
| Superpixel | 211.3 | **26.23** | 140.79 | **17.9** |

# Результаты, выносимые на защиту

- Предложен метод увеличения эффективности нейронных сетей в задаче распознования символов, основанный на упрощении входных данных.
- Показана эффективность предложенного метода в терминах используемой во время обучения памяти при осутствии существенного уменьшения качества по сравнению со стандартным подходом.
- Проведено сравнение времени обучения и предсказания нейросетей, работающих со скелетами и суперпикселями соответсвенно.