

Выделение контуров в графических
изображениях с использованием
нейросетевой технологии

Милевски Александр

Оглавление

1	Обзор существующих детекторов контуров в графических изображениях	3
2	Введение в теорию нейронных сетей	4
3	Применение нейросетевой технологии для выделения контуров	5
3.1	Введение	5
3.2	Нейронная сеть для определения характеристик границ . . .	6
3.2.1	Взвешенный по входу нейрон-модель	7
3.2.2	Расчёт выхода подсети	8
3.2.3	Описание и выявление контуров	10
3.3	Топология сети	11
3.3.1	Представление информации о контурах	11
3.3.2	Определение подсети-победителя U_{rwin}	12
3.3.3	Определение нейрона-победителя $V_{rwin swin}$ в подсети U_r	13
3.3.4	Нейрон динамического отслеживания V_d	14
3.3.5	Двоичное представление контура	15
3.3.6	Сравнение с топологией классической нейронной сети, основанной на моделях	16
3.4	Обучение	17
3.4.1	Выбор p_{rwin} подсети U_{rwin} и её обучение	17
3.4.2	Изменение $w_{rwin swin}$ для нейрона $V_{rwin swin}$	18
3.4.3	Получение подходящих конфигураций контуров	18
3.5	Выделение контуров	19
3.5.1	Выявление первичных потенциальных контурных точек	19
3.5.2	Выявление вторичных контурных точек	20
3.6	Схема работы контурного фильтра	21
3.6.1	Входные данные	21

3.6.2	Процесс выявления контуров	21
3.6.3	Обучение	23
4	Программная реализация	25
5	Исследование эффективности разработанной системы	26

Глава 1

Обзор существующих детекторов контуров в графических изображениях

Глава 2

Введение в теорию нейронных сетей

Глава 3

Применение нейросетевой технологии для выделения контуров

3.1 Введение

Выявление контуров, помимо описанных фильтров, можно осуществлять с помощью нейросетевой технологии. Используя подборку карт контуров в качестве базы данных для обучения сети специально спроектированной архитектуры, можно получить сеть, выполняющую задачу выделения контуров и способную абстрагироваться от изображений обучающего набора и выполнять поставленную задачу на изображениях, не входивших в базу данных.

В общем случае, задача выделения контуров состоит из двух стадий: выявление участков изображения, существенно отличающихся по яркости от соседних с ними, и сравнение выявленных отличий с неким предзаданным пороговым значением, определяющим действительную достаточность этих различий в участках для объявления их участками контура. На первой стадии требуется аппарат выявления отличий в яркости, в роли которого может выступать, к примеру, оператор Прюитта, Собеля или Робертса. На второй же стадии, детектору обычно требуется задать пороговые значения для оптимальной обработки изображения.

Выделение контуров грубыми способами является интерактивным процессом, в котором пользователь должен корректировать глобальные пороговые параметры, так как результат редко сразу бывает удовлетворительным по той причине, что не настроенный под конкретное изображение детектор может ошибочно выделять зашумлённые участки изоб-

ражения, как содержащие контурные точки. Более сложные методы, такие как детектор Кенни и детектор Шена-Кастана, используют так называемую операцию порогового гистерезиса, при которой за значимый контур принимается последовательность пикселей, в которой отличие в яркости от окружения по меньшей мере одного пикселя превосходит верхнее пороговое значение, тогда как остальные пиксели последовательности превышают нижний порог. К тому же, для выявления характеристик контура на каждом участке шириной в один пиксель требует использования преобразования Лапласа от фильтра Гаусса для выделения нулевых пересечений, в связи с чем требуется задание дополнительных параметров. Добавление этих параметров к уже имеющимся пороговым увеличивает количество вариаций наборов параметров, каждый из которых даст свою контурную карту изображения.

Преимущество применения нейросетевой технологии в данной задаче заключается в том, что оно позволяет сократить количество параметров детектора.

Учитывая, что сложная операция выявления контуров тем не менее потребует определения достаточно большого количества параметров, логично будет представить их в виде весов связей в нейронной сети. За основу следует взять иерархическую архитектуру, предложенную Кунгом и Тауrom и изменить её для решения задачи выявления контуров таким образом, что веса связей будут играть двойную роль - моделировать разновидности окружения контуров на начальной стадии поиска контуров и выступать в качестве пороговых параметров на финальной стадии принятия решения.

3.2 Нейронная сеть для определения характеристик границ

Обычно иерархическая нейронная сеть состоит из набора взвешенных, или параметризованных по весу нейронов-моделей в качестве вычислительных элементов, а выход отдельной подсети определяется линейной комбинацией локальных выходов входящих в неё нейронов. В случае задачи поиска контуров эта схема нуждается в модификации: параметризованные по весу нейроны будут заменены альтернативной моделью - так называемыми взвешенными по входу нейронами, а выход подсети будет определяться в ходе соревновательного процесса по принципу "победитель получает всё".

3.2.1 Взвешенный по входу нейрон-модель

Вместо отображения вложенного вектора весов $z \in R^M$ меньшей размерности в весовой вектор высокой размерности $p \in R^N$ (как это делается в случае параметризованного по весу нейрона), в схеме с взвешенным по входу нейроном: вектор входов высокой размерности $x \in R^N$ отображается в вектор низкой размерности $x^P \in R^M$. Это делается из соображений, что вектор может исчерпывающе описать наиболее важные характеристики своего прообраза более высокой размерности x . Если такое отображение существует для множества векторов x , то векторы весов можно использовать в их форме z меньшей размерности в сети вместо исходной формы вектора высокой размерности p .

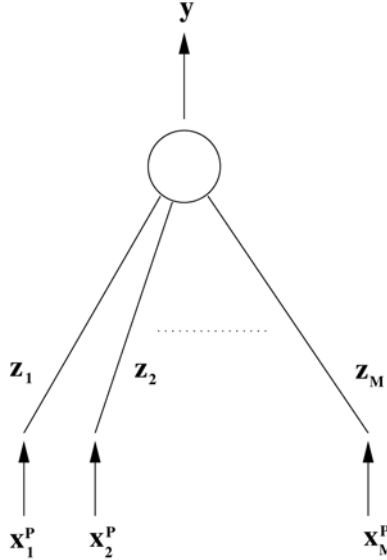


Рис. 3.1: Взвешенный по входу нейрон

Предположим, что существует отображение $\mathcal{P}: R^N \longrightarrow R^M$ такое, что $x^P = \rho(x) \in R^M$, где $M < N$. Операцию такого нейрона-модели можно определить следующим образом:

$$y_s = f_s(x, p) = f_s(x^P, z) \quad (3.1)$$

Схема параметризованного по входу нейрона показана на Рис.3.1.

Оба описанные выше типа нейронов позволяют проводить подбор оптимальных весов для увеличения эффективности работы в пространстве

меньшей размерности. В выборе вида нейрона следует исходить из существования отображения \mathcal{M} для весовых коэффициентов или \mathcal{P} для входов: если пространство параметров ограничено подпространством меньшей размерности в пространстве высокой размерности, следует применять параметризованный по входу нейрон. Его использование также имеет смысл даже в случае, когда такое представление вектора весовых коэффициентов заранее не известно, а из распределения входного вектора или после применения метода главных компонент обнаруживаются признаки существования подходящего оператора с минимальной потерей информации, который производит отображение входного вектора в пространство меньшей размерности. Именно так дело обстоит с нейронной сетью для поиска контуров, которая производит отображение участка пикселей в приконтурном участке в векторы размерности 2, отражающие два доминирующих уровня яркости вокруг контура.

3.2.2 Расчёт выхода подсети

В выбранном подходе для расчёта выхода подсети вместо линейной комбинации выходов входящих в неё нейронов используется соревновательный процесс, победитель в котором определяется следующим образом:

$$p_{win} = \underset{s}{\operatorname{argmax}} \underset{r}{(x, p_s)} \quad (3.2)$$

где p_{win} - индекс нейрона-победителя.

Выход подсети $\phi(x, p_r)$ заменяется выходом нейрона-победителя.

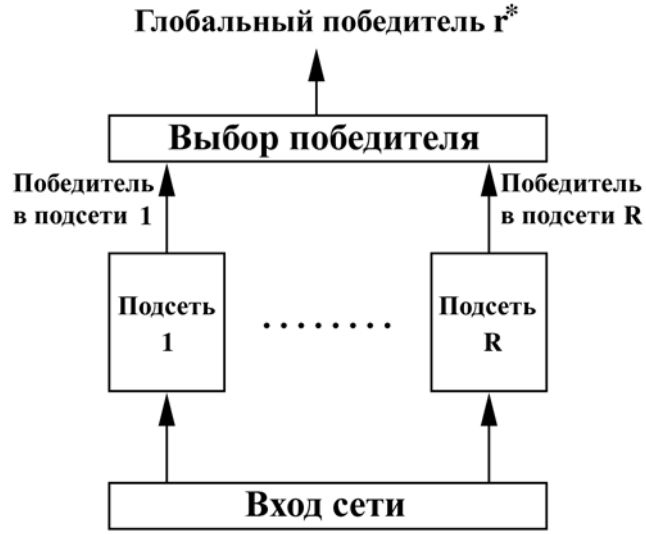
$$\phi(x, p_r) = \phi(x, p_{win}) \quad (3.3)$$

С учётом этого соревновательного процесса, для вычисления выхода отдельного нейрона естественно использовать Эвклидово расстояние

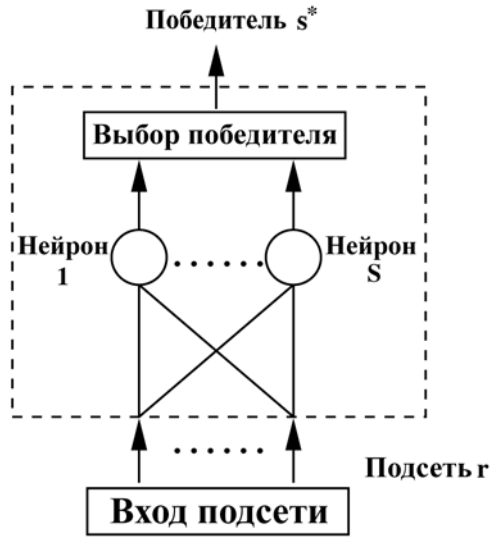
$$r(x, p_{win}) = \|x - p_{win}\| \quad (3.4)$$

Такой тип сетей хорошо подходит для задач классификации образов без наблюдателя, при которой каждый класс образов состоит из нескольких разобщённых наборов слабо отличающихся характеристик. Тогда каждый главный, первичный класс можно связать с подсетью, каждый производный внутри него - с нейроном этой подсети. Полученная топология сети и структура отдельной подсети показана на Рис.3.2

В взвешенном по входу нейроне не производится включения вектора z_s в пространство более высокой размерности, а происходит отображение



(a)



(b)

Рис. 3.2: Топология нейронной сети для выделения контуров: (a) архитектура сети, (b) архитектура подсети

входного вектора $x \in R^N$ в подпространство меньшей размерности R^M с помощью оператора \mathcal{P} . Таким образом:

$${}_r(x, p_s) \equiv_r (x^P, z_s) \quad (3.5)$$

$$= {}_r(\mathcal{P}(x), z_s) \quad (3.6)$$

где $x^P = \mathcal{P}(x)$ - входной вектор пониженной размерности, то есть образ входного вектора x в R^N .

3.2.3 Описание и выявление контуров

Выбор данной топологии сети основан на наблюдении различных предпочтений человека в отношении определённой величины различия между уровнями яркости в качестве границы при разных условиях освещённости. Чтобы включить этот критерий в процесс выделения контуров, целесообразнее использовать предложенную иерархическую топологию сети, в которой для представления различных уровней освещённости предназначены подсети, а нейроны внутри них - для выполнения роль представления различных прототипов контуров при заданном подсетью уровне освещённости. Под *прототипом контура* понимается вектор размерности 2 $w \in R^2$ отражающий два доминирующих уровня яркости по обе стороны контура.

Так как прототипы создаются в процессе обучения на основе отобранных человеком образцов, следует применить схему состязательного обучения без наблюдателя, а каждый прототип представить в сети при помощи вектора весовых коэффициентов нейрона. Соревнование по правилу "победитель получает всё" также располагает к использованию иерархической топологии с подклассами, внутри которых во время обучения только нейрон-победитель может обновлять свой вектор весов. Необходимо добавить, что выход нейрона внутри r -той подсети вычисляется как Евклидово расстояние между прототипом и образцом участка контура.

$$r(x^P, z_{s_r}) = \|x^P - z_{s_r}\| \quad (3.7)$$

где $x^P \in (R)^2$ текущий обрабатываемый участок, а z_{s_r} - s_r -тый прототип границы в рамках r -той подсети.

С учётом того факта, что отдельный участок контура обычно представлен набором значений уровня яркости $x \in (R)^N$, где $N \gg 2$, необходимо суммировать эти значения для выделения двух преобладающих уровней яркости. Иными словами, необходимо получить отображение $\mathcal{P}: R^N \rightarrow R^2$ такое, что

$$x^P = \mathcal{P}(x) \quad (3.8)$$

что будет соответствовать операции взвешенного по входу нейрона-модели.

3.3 Топология сети

Предложенная топология нейронной сети состоит из некоторого количества подсетей, каждый нейрон которых ответственен за усвоение определённого набора образцов из базы данных обучения сети. На начальной стадии обучения образцы обучающего набора адаптивно разделяются на подмножества в ходе не управляемого извне соревновательного процесса между подсетями. В последующей стадии - распознавании - пиксели контуров выявляются путём сравнения конфигурации текущего пиксела с прототипами, связанными с разными подсетями.

Применение иерархической топологии основано на наблюдении, что при выделении контуров более эффективнее использовать множество наборов пороговых параметров для принятия решения при различных условиях окружения, чем использовать один набор параметров для всего изображения, как это делается в других методах.

В предложенной модели представления каждая подсеть связана с шаблоном контура при определённом уровне фоновой освещённости, а каждый нейрон подсети соответствует одной из возможных вариаций контура при данной освещённости. Топология этой классифицирующей сети-детектора показана на Рис3.3, а иерархическая схема представления прототипов описана в следующих разделах.

3.3.1 Представление информации о контурах

Выделение контуров проводится на $N \times N$ -окрестности текущего пиксела. При конкатенации соответствующих значений уровней яркости в вектор $x = [x_1 \dots x_{N^2}]^T \in R^N$ средний уровень яркости рассчитывается следующим образом:

$$\bar{x} = \frac{1}{N^2} \sum_{n=1}^{N^2} x_n \quad (3.9)$$

После задания этого среднего значения информация об уровнях яркости обрабатываемого участка дальше может быть представлена в виде вектора $m = [m_1 m_2]^T \in R^2$, компоненты которого соответствуют двум преобладающим уровням яркости внутри участка и определяются так:

$$m_1 = \frac{\sum_{i=1}^{N^2} I(x_i < \bar{x}) x_i}{\sum_{i=1}^{N^2} I(x_i < \bar{x})} \quad (3.10)$$

$$m_2 = \frac{\sum_{i=1}^{N^2} I(x_i \geq \bar{x}) x_i}{\sum_{i=1}^{N^2} I(x_i \geq \bar{x})} \quad (3.11)$$

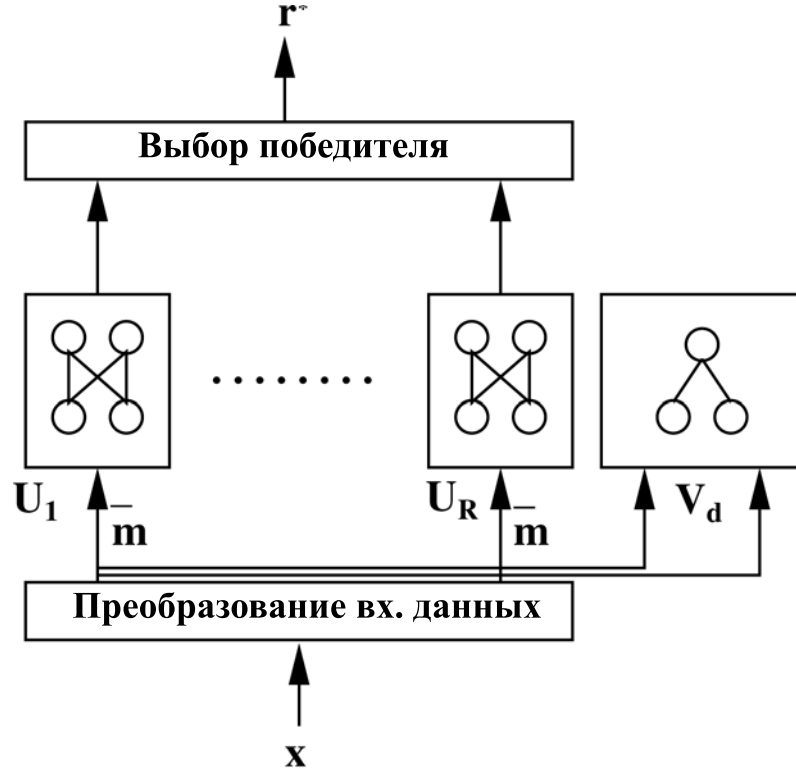


Рис. 3.3: Нейронная сеть для выделения контуров

$$\overline{m} = \frac{m_1 + m_2}{2} \quad (3.12)$$

где функция $I(\bullet)$ является функцией истинности, обращаясь в единицу, если её аргумент несёт логически истинное значение, и в ноль в ином случае.

3.3.2 Определение подсети-победителя $U_{r_{win}}$

Как было описано ранее, каждая подсеть $U_r, r = 1, \dots, R$ связана с прототипом значения уровня яркости фоновой освещённости p_r . Локальные участки $N \times N$ в изображении со средними значениями уровня яркости, ближайшими к p_r отправляются для дальнейшей обработки в подсеть U_r . Если говорить подробнее, участок пикселей \mathcal{W} в изображении с его средними показателями яркости \overline{m}, m_1, m_2 , полученными из уравнений с (3.10) по (3.12), присваивается подсети $U_{r_{win}}$, если выполняются следующие условия:

$$p_{r_{win}} \in [m_1, m_2] \quad (3.13)$$

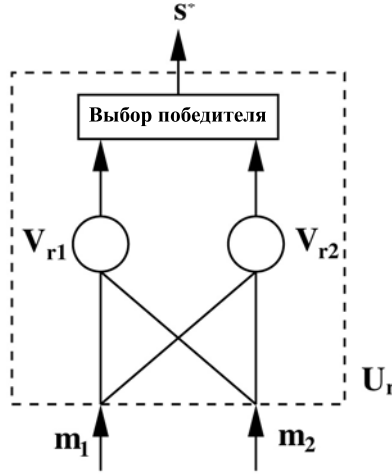


Рис. 3.4: Схема подсети u_R

$$|\bar{m} - p_{r_{win}}| < |\bar{m} - p_r| \quad r = 1, \dots, R, r \neq r_{win} \quad (3.14)$$

где $[m_1, m_2]$ - закрытый интервал с границами m_1, m_2 . Множество выделенных участков $N \times N$, таким образом, разбивается на подмножества, элементы которых представляют собой различные уровни освещённости. Для удобства обозначим два условия (3.13) и (3.14) вместе как $x \rightarrow U_{r_{win}}$. Архитектура подсети показана на Рис.3.4.

3.3.3 Определение нейрона-победителя $V_{r_{win}s_{win}}$ в подсети U_r

Каждая подсеть U_r состоит из S нейронов $V_{rs}, s = 1, \dots, S$, которые являются шаблонами контуров, представляя возможные вариации контуров при среднем общем уровне освещённости p_r . Каждый нейрон связан с весовым вектором $w_{rs} = [w_{rs,1} w_{rs,2}]^T \in R^2$, который объединяет два преобладающих уровня яркости в каждом участке $N \times N$ пикселей \mathcal{W} в форме вектора-прототипа m . Участок с принадлежащим ему вектором m присваивается нейрону $V_{r_{win}s_{win}}$, если выполняется следующее условие:

$$\|m - w_{r_{win}s_{win}}\| < \|m - w_{rs}\| \quad s = 1, \dots, S, s \neq s_{win} \quad (3.15)$$

При S , равном 2, каждая подсеть состоит из двух нейронов, из которых один представляет собой прототип слабо выраженного контура, другой - прототип явно выраженного контура. Соответственно, один из

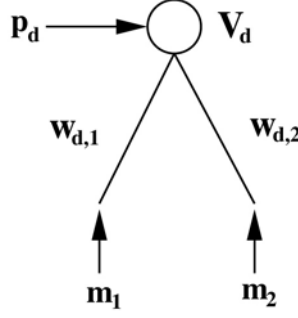


Рис. 3.5: Нейрон динамического отслеживания контура

весовых векторов $w_{r_{win}s}$, $s = 1, 2$ становится прототипом слабого контура $w_{r_{win}}^l$, а второй - прототипом сильного контура $w_{r_{win}}^u$. К какому виду относится нейрон, определяется согласно следующему критерию:

$$\begin{aligned} w_{r_{win}}^l &= w_{rs'}, s' = \operatorname{argmin}_s (w_{r_{win}s,2} - w_{r_{win}s,1}) \\ w_{r_{win}}^u &= w_{rs''}, s'' = \operatorname{argmax}_s (w_{r_{win}s,2} - w_{r_{win}s,1}) \end{aligned}$$

После назначения прототипу слабого контура весового вектора $w_{r_{win}}^l = [w_{r_{win},1}^l w_{r_{win},2}^l]^T$, мера $(w_{r_{win},2}^l - w_{r_{win},1}^l)$ играет роль порогового параметра в общепринятых алгоритмах выделения контура в определении нижней границы видимости контура и полезна для нахождения потенциальных начальных точек контура в изображении для его отслеживания. Структура нейрона показана на Рис.3.5.

3.3.4 Нейрон динамического отслеживания V_d

В дополнение к подсетям U_r и локальным нейронам V_{rs} в топологию сети вводится *нейрон динамического отслеживания контура* V_d . Другими словами, этот нейрон находится вне подсетей, в глобальном пространстве сети.

Динамический нейрон является гибридом подсети и отдельного нейрона, который содержит и динамический весовой вектор $w_d = [w_{d,1} w_{d,2}]^T \in R^2$, соответствующий весовому вектору отдельного нейрона, и скалярный параметр p_d , аналогичный индикатору уровня освещённости подсети. Структура динамического нейрона отслеживания показана на Рис.(3.5).

Задачей этого нейрона является слежение за изменением уровня яркости фона во время отслеживания контура. На стадии обучения нейрон

неактивен. На стадии распознавания и выделения явных точек контура *динамический весовой вектор* $w_d = [w_{d,1} w_{d,2}]^T \in R^2$ и индикатор уровня освещения фона p_d постоянно изменяются, чтобы отследить менее явные точки контура, связанные с начальной его точкой, то есть динамический нейрон непосредственно проходит стадию обучения в процессе обработки участка контура в изображении.

3.3.5 Двоичное представление контура

Предположим, что вектор m текущего участка \mathcal{W} привязан к нейрону $V_{r_{win} s_{win}}$ с весовым вектором $w_{r_{win} s_{win}}$. Для выделения контуров целочисленный вектор $x \in R^{N^2}$ представляет значения уровней яркости пикселей в текущем участке и отображается в двоичный вектор $b \in B^{N^2}$, где $B = \{0, 1\}$. Чтобы добиться этого, определим операцию отображения $\mathcal{Q}: R^{N^2} \times R^2 \rightarrow B^{N^2}$ следующим образом:

$$b = \mathcal{Q}(x, w_{r_{win} s_{win}}) = [q(x_1, w_{r_{win} s_{win}}) \quad \dots \quad q(x_{N^2}, w_{r_{win} s_{win}})]^T \in B^{N^2} \quad (3.16)$$

где входящие в него отображения $q: R \times R^2 \rightarrow B$ заданы как

$$q(x_n, w_{r_{win} s_{win}}) = \begin{cases} 0 & \text{if } |x_n - w_{r_{win} s_{win},1}| < |x_n - w_{r_{win} s_{win},2}| \\ 1 & \text{if } |x_n - w_{r_{win} s_{win},1}| \geq |x_n - w_{r_{win} s_{win},2}| \end{cases} \quad (3.17)$$

Для верных конфигураций контура бинарные векторы b принимают фиксированные формы, показанные на Рис.(3.6) для $N = 3$. На начальной стадии обучения конфигурация контура, связанная с каждым образом контура, записанная в двоичной форме, хранится в *наборе конфигураций контуров* C , который является частью набора параметров самой сети. Множество в дальнейшем расширяется, потому что при $N = 3$ оно замкнуто относительно операции $R_{\frac{\pi}{4}}$. В частности, эта операция является такой перестановкой над вектором $b \in B^9$, что, будучи представленной на матрице размером 3×3 , является поворотом на 45° по часовой стрелке b . Это показано на иллюстрации Рис.3.7. Таким образом, выделение центральных поворотов конфигураций контуров, которые не были представлены на стадии обучения, облегчено.

0	0	0
0	0	0
1	1	1

0	0	1
0	0	1
0	0	1

0	1	1
0	0	1
0	0	0

Рис. 3.6: Примеры конфигураций краёв

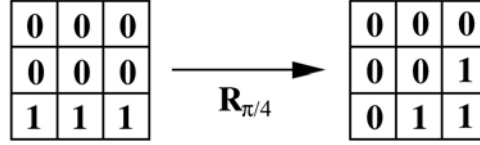


Рис. 3.7: Операция $R_{\pi/4}$

3.3.6 Сравнение с топологией классической нейронной сети, основанной на моделях

Представленная нейронная сеть, выделяющая контуры, находится в прямом соответствии с обобщённой топологией нейронной сети, основанной на моделях, которое состоит в следующем: подсети $U_r, r = 1, \dots, R$ тождественны подсетям в классической модели, а нейроны $V_{rs_r}, s_r = 1, \dots, S_r$ соответствуют нейронам, входящим в подсети классической архитектуры, в данном случае, числом $S_r = 2$ на каждую подсеть.

Вместо включения весового вектора малой размерности в пространство высокой размерности, как делается в случае с параметризованным по весу нейроном, вектор входных данных относительно высокой размерности отображается в вектор меньшей размерности m , содержащий информацию о двух доминирующих уровнях яркости в рассматриваемом участке. Другими словами, использовался взвешенный по входным данным нейрон-модель, описанный раньше, с функцией отображения $\mathcal{P}: R^{N^2} \rightarrow R^2$, заданной таким образом, что

$$m = \mathcal{P}(x) \quad (3.18)$$

Наряду с изложенными выше соответствиями, есть ряд небольших различий между исходной топологией классической нейронной сети с иерархической архитектурой с подразделами и представленной сетью, выделяющей контуры. В исходной архитектуре выходные данные подсети полностью заменяются выходными данными принадлежащего ей нейрона-победителя следующим образом:

$$\phi(x, w_r) \equiv_r (x, w_{s_{win}}) \quad (3.19)$$

где s - индекс локального победителя, а $r(x, w_{s_{win}})$ его выходные данные. В данной сети соревновательный процесс на уровне подсетей независим от соревновательного процесса на уровне отдельных нейронов и имеет совсем другую природу: соревнование между подсетями происходит согласно соответствию в показателях текущего локального уровня яркости освещённости фона и уровня освещённости фона прототипа, что обозначает сравнение скалярных значений. Соревнование же на уровне нейронов состоит в сравнении векторов размерности 2 - прототипов контуров при конкретном уровне яркости фона. В результате, выходные данные подсетей и нейронов независимы и иерархически находятся на двух разных уровнях. На уровне подсетей выход формулируется через показатели яркости фона:

$$\phi(\bar{m}, p_r) = |\bar{m} - p_r| \quad (3.20)$$

На уровне нейронов локальный выход формулируются через векторы-прототипы контуров:

$$r(m, w_{rs}) = \|\bar{m} - w_{rs}\| \quad (3.21)$$

3.4 Обучение

Обучение состоит из трёх стадий: на первой стадии в соревновательном процессе задаются прототипы средней фоновой освещённости p_r , $r = 1, \dots, R$ для каждой подсети U_r . На второй стадии каждый участок \mathcal{W} приписывается своей подсети, и локальному нейрону этой подсети на основании его параметров p_r и m . Весовой вектор w_{rs} нейрона-победителя настраивается в соревновательном процессе обучения. На третьей стадии на основании отображения \mathcal{Q} формируется соответствующая конфигурация двоичного образа b как функция весового вектора победителя w_{rs} . Далее к b применяется операция $R_{\frac{\pi}{4}}$ к b , чтобы получить восемь центральных поворотов образа, и сохраняем их в набор конфигураций \mathcal{C} . Размер стороны обрабатываемого участка принимается $N = 3$,

3.4.1 Выбор $p_{r_{win}}$ подсети $U_{r_{win}}$ и её обучение

Допустим, что текущий обрабатываемый участок с его средним показателем p_r приписан подсети $U_{r_{win}}$ на основании условий (3.13) и (3.14).

Затем значение $p_{r_{win}}$ модифицируется в соревновательном обучении следующим образом:

$$p_{r_{win}}(t+1) = p_{r_{win}}(t) + \eta(t)(\overline{m} - p_{r_{win}}(t)) \quad (3.22)$$

Корректирующий множитель $\eta(t)$ последовательно уменьшается по следующему правилу:

$$\eta(t+1) = \eta(0) \left(1 - \frac{t}{t_f}\right) \quad (3.23)$$

где t_f - общее количество итераций обучения.

3.4.2 Изменение $w_{r_{win}s_{win}}$ для нейрона $V_{r_{win}s_{win}}$

Допустим, что настоящий участок изображения с его вектором параметров m приписан локальному нейрону $V_{r_{win}s_{win}}$ подсети $U_{r_{win}}$, его весовой вектор $w_{r_{win}s_{win}}$ нейрона снова корректируется в соревновательном обучении следующим образом:

$$w_{r_{win}s_{win}}(t+1) = w_{r_{win}s_{win}}(t) + \eta(t)(m - w_{r_{win}s_{win}}(t)) \quad (3.24)$$

причём множитель $\eta(t)$ последовательно уменьшается согласно уравнению (3.23).

3.4.3 Получение подходящих конфигураций контуров

По окончании предыдущих стадий, индикаторы фонового уровня освещённости p_r подсетей и весовые векторы w_{rs} для нейронов определены. В результате, все участки изображения размером $N \times N$ могут быть распределены по подходящим подсетям и соответствующим нейронам подсетей согласно их параметрам \overline{m} и m . Если текущий участок приписан нейрону $V_{r_{win}s_{win}}$ внутри подсети $U_{r_{win}}$, вектор уровней яркости x этого участка \mathcal{W} может быть переведён в бинарный вектор b конфигурации контура как функция $w_{r_{win}s_{win}}$ согласно уравнению (3.17):

$$b = \mathcal{Q}(x, w_{r_{win}s_{win}}) \quad (3.25)$$

Двоичный вектор b добавляется во множество допустимых конфигураций контура \mathcal{C} . При работе с размером стороны участка $N = 3$, условие замкнутости множества по операции $R_{\frac{\pi}{4}}$ может быть выполнено, если сгенерировать восемь конфигураций контура $b_j, j = 0, \dots, 7$, используя операцию поворота $R_{\frac{\pi}{4}}$ следующим образом:

$$b_0 = b \quad (3.26)$$

$$b_{j+1} = R_{\frac{\pi}{4}}(b_j) \quad j = 0, \dots, 6 \quad (3.27)$$

и сохраняя их в множестве конфигураций C .

3.5 Выделение контуров

На этой стадии все $N \times N$ участки тестового изображения проверяются на наличие характеристик контура. Эта стадия распознавания состоит из двух этапов. На первом этапе все пиксели с высокой степенью сходства с усвоенными сетью прототипами контуров объявляются наиболее вероятными точками контуров. На втором этапе эти наиболее вероятные контурные точки используются, как начальные точки для операции отслеживания контура, которая рекурсивно проверяет менее явные контурные точки, определённые как вторичные.

3.5.1 Выявление первичных потенциальных контурных точек

На этом этапе все участки $N \times N$ тестового изображения \mathcal{W} подвергаются проверке. Все участки, параметры которых \overline{m}, m_1, m_2 удовлетворяют следующим условиям, объявляются первичными контурными точками:

- (A1). $x \rightarrow U_{r_{win}}$ выполнение условий (3.13) и (3.14)
- (A2). $m_2 - m_1 \geq w_{r_{win},2}^l - w_{r_{win},1}^l$, где $w_{r_{win}}^l$ - вектор-прототип слабо видимого контура $U_{r_{win}}$
- (A3). $b = \mathcal{Q}(x, w_{r_{win}s_{win}}) \in C$, где $w_{r_{win}s_{win}}$ - весовой вектор, связанный с нейроном $V_{r_{win}s_{win}}$

Условие (A1) означает, что средний уровень яркости текущего участка должен быть близок к показателю одной из запроецированных подсетей. Условие (A2) подтверждает, что значительность границы, выраженная разницей между m_2 и m_1 , больше, чем разница между компонентами прототипа слабо видимого контура. Условие (A3) проверяет что двоичный образ контура данного участка - одна из допустимых конфигураций множества .

3.5.2 Выявление вторичных контурных точек

На втором этапе активируется нейрон динамического отслеживания контура V_d , чтобы отследить вторичные контурные точки, связанные с текущей первичной точкой. Индикатор уровня яркости p_d и весовой вектор w_d нейрона инициализируются при помощи параметров обнаруженной первичной контурной точки \overline{m}^p и $m^p = [m_1^p \ m_2^p]^T$ следующим образом:

$$p_d(0) = \overline{m}^p \quad (3.28)$$

$$w_d(0) = m^p \quad (3.29)$$

После инициализации параметров динамического нейрона начинает работу рекурсивный алгоритм отслеживания контура, чтобы найти менее явно выраженные пиксели контура (вторичные точки), связанные с первичными точками контура, применяя следующий набор условий к каждому из 8 соседних участков выбранной первичной точки.

$$(B1). \quad b = \mathcal{Q}(x, w_d) \in C$$

$$(B2). \quad p_d \in [m_1, m_2]$$

Условие (B1) похоже на условие (A3) для распознавания первостепенных контурных точек, в то время, как условие (B2) является модификацией условия (3.13), составляющего часть набора необходимых условий для $x \rightarrow U_r$, подтверждающих, что средний уровень яркости потенциальной вторичной точки контура, представленный параметром p_d , близок к таковому предыдущей обработанной точки. В дополнение к этому, никаких больше условий на силу контура в текущем пикселе не налагается, чтобы позволить включить в окончательный контур слабо видимые участки, если они связаны с более явными точками контура.

Для каждой новой вторичной точки контура с соответствующими параметрами \overline{m}^s и m^s , удовлетворяющей условиям, описанным выше, индикатор локального уровня яркости p_d динамического нейрона корректируется следующим образом:

$$p_d(t+1) = p_d(t) + \eta(t)(\overline{m}^s - p_d(t)) \quad (3.30)$$

В дополнение к этому, если точка контура удовлетворяет условию (A2) детекции первичных точек контура, показывающему, что её значимость сопоставима с значимостью явной первичной точки контура, весовой вектор w_d динамического нейрона изменяется для регистрации характеристик текущей точки:

$$w_d(t+1) = w_d(t) + \eta(t)(m^s - w_d(t)) \quad (3.31)$$

3.6 Схема работы контурного фильтра

Работу контурного фильтра на основе разработанной нейронной сети можно алгоритмически описать в виде автомата, на вход которого поступают графические изображения, а результатом его работы являются карты контуров объектов на этих изображениях. Обучение нейронной сети представляет собой этап начальной калибровки автомата при помощи базы специально отобранных образцов эталонов, которая производится однократно и не требует повторения перед каждым циклом работы.

3.6.1 Входные данные

Алгоритм работы фильтра не зависит от формата, цветовой палитры и размеров предлагаемых для обработки изображений, так как фильтр обрабатывает изображения дискретно, разбивая их на равные участки и используя только яркостную составляющую палитры изображений. Параметризация входных данных состоит в сокращении информации об участке от матрицы уровней яркости входящих в него пикселей до набора трех характеристических значений - среднего уровня яркости участка \overline{m} и средних верхних и нижних значений яркости пикселей участка m_1 и m_2 , получаемых из формул (3.10)-(3.12). Далее соревновательные процессы в сети будут зависеть только от функций этих значений.

3.6.2 Процесс выявления контуров

После разбиения обработанного входного изображения на участки, стадия выявления первичных участков контура состоит в том, что каждый участок с его рассчитанными характеристиками проходит полный путь по графу сети последовательно через слой подсетей, где выбор соответствующей характеристикам участка подсети представляет собой классификацию участка по среднему уровню освещённости, к слою нейронов выбранной подсети, где выбор победителя равнозначен принятию решения о значимости обнаруженных внутри участка различий в уровнях яркости. Этап соревнования подсетей моделирует поведение человека в решении задачи выявления границ объектов - при различных средних уровнях освещённости оценка значимости контура происходит с разными параметрами.

Как было объяснено выше, после выбора соответствующей участку подсети соревновательный процесс для принятия решения о контуре переносится на уровень пары нейронов подсети-победителя, чтобы в дальнейшем выход нейрона-победителя заменил выход подсети, к которой он

принадлежит, в соответствии со схемой "победитель получает всё". Нейроны подсети представляют собой модели контуров при данном освещении, причём один моделирует явные различия в яркости внутри участка, то есть контуры объектов, нахождение которых и является сутью поставленной задачи, а другой - слабые различия, то есть те участки, на которых на этой стадии контур выявлен не будет (см. условие (A2)).

Выявление первичных участков контура

Таким образом выходом нейрона-победителя, а, следовательно, и подсети, которой он принадлежит, становится либо карта контура полученная преобразованием Q над значениями яркости пикселей участка, либо пустая карта, в случае, если победителем оказался нейрон, моделирующий слабые различия в яркости в участке. На выход нейрона-победителя, если таковым оказался нейрон, моделирующий явные различия при данном уровне освещённости, налагается ограничение соответствия полученной карты контура одной из конфигураций контура, содержащейся в наборе возможных конфигураций контуров нейронной сети, сформированном в процессе обучения. Если это условие выполняется, контурная карта участка заносится в карту контуров изображения, частью которого он является.

Выявление вторичных участков контура

Как было показано в предыдущем разделе, участки изображения, внутри которых различия в яркости пикселей при их среднем уровне яркости принимаются за незначительные, не дают карты контура на выходе нейрона, к которому они были приписаны. Но предложенная модель сети способна учитывать то, что эти участки могут быть смежными с участками, на которых контур выявляется явно, и содержать его продолжение, яркость которого снижена изменившимся уровнем освещённости или же связанной с ним потерей информации при сжатии изображения. За выявление контура в таких ситуациях отвечает нейрон динамического отслеживания V_d , обрабатывающий окрестности участков, в которых на предыдущей стадии был обнаружен контур объекта. Таким образом, по окончании стадии выявления первичных участков контура, наборы из 8 участков, окружающих участки изображения, на которых был выделен контур, проходят вторичную обработку с помощью нейрона V_d , стоящему в топологии сети вне всех подсетей и совмещающего в себе одновременно моделирующие свойства как нейрона, так и подсети. Нейрон динамического отслеживания контура инициализирует свои параметры

на основании характеристик участка, окрестность которого ему предстоит обработать - среднего общего уровня яркости \overline{m} и средних верхнего и нижнего уровней яркости m_1 и m_2 . Далее на участках окрестности он проводит операцию выявления контура с менее строгим набором условий (B1), (B2), совмещённую с коротким циклом обучения последовательно корректирующим его прототип среднего значения яркости участка, а если в окрестности встречается область с признаками первичного участка контура, то динамический нейрон корректирует также свои пороговые значения, от которых зависит результирующая в обрабатываемых участках карта контура.

Результат

По завершении обеих стадий результатом обработки является изображение, соответствующее по размеру представленному на входе, монохромное, где контуры объектов, выявленные фильтром, выделены выбранным цветом. Для корректности разбиения на участки входное изображение, в случае необходимости, должно быть увеличено, чтобы его размеры были кратны размерам участка, и результирующее изображение потребует обратного преобразования, что следует выполнить до перевода матрицы значений яркости в графический формат.

3.6.3 Обучение

Обучение сети в последовательности действий равноценно многократной обработке входного изображения сетью с предустановленными на начальном цикле параметрами её подсетей и нейронов, с тем лишь схематическим отличием от операции выявления контура, что в нём не участвует динамический нейрон, и, следовательно, распознавание проходит в одну стадию. При этом каждый раз, когда выбирается победитель среди подсетей или нейронов, их параметры корректируются согласно (3.22) и (3.24), достигая, таким образом, оптимальных значений в зависимости от базы данных обучения. Результаты обработки сравниваются наблюдателем с образцами контуров вплоть до получения удовлетворительной оценки. Чтобы сеть сформировала способность абстрагироваться от обучающего набора изображений, следует обучать сеть с помощью наибольшего возможного количества различных по своим характеристикам изображений, что является трудоёмкой задачей. Полученный в результате обучения набор параметров элементов сети следует сохранить и использовать для инициализации параметров сети перед началом её работы.

Предустановку параметров подсетей перед обучением предполагается делать следующим образом: точки на шкале серого, соответствующие прототипам уровней яркости подсетей расположить равномерно от начала до конца шкалы, установив количество подсетей в соответствии с желаемым уровнем подробности выявления контуров и располагаемыми вычислительными ресурсами, которые будут использоваться фильтром. Пороговые параметры нейронов следует установить нулевыми, и они будут инициализированы набором значений первого приписанного нейрону обрабатываемого участка изображения согласно выражению (3.23) для корректирующего множителя. Если по окончании обучения в сети останутся элементы, ни разу не задействованные в процессе обучения, их целесообразнее будет удалить, так как их использование будет сопряжено с потерей машинного времени и ресурсов, не давая при этом желаемого и предсказуемого результата, или же они будут усложнять принятие решения в соревновательном процессе, участвуя в нём лишь номинально.

Глава 4

Программная реализация

Глава 5

Исследование эффективности разработанной системы