

# **Предварительная обработка в системе видеонаблюдения на основе искусственных нейронных сетей**

## **1. Введение**

Процесс предварительной обработки данных - важный этап при анализе видеоданных с помощью искусственных нейронных сетей (ИНС). Известно, что практически все виды нейронных сетей очень чувствительны к различным смещениям и деформациям при распознавании объектов. Это связано с тем, что нейронная сеть сравнивает изображение в памяти с текущим поэлементно, при этом важно точное позиционирование текущего объекта по отношению к ранее запомненному. Если такое позиционирование не будет выполнено, то для нейронной сети два сравниваемых объекта будут существенно отличаться.

Данную проблему пытаются решить различными способами, в частности, наиболее известны два способа:

1. Предъявить при обучении заранее подобранные изображения всевозможных вариаций (различные смещения, уровни освещенности), например, при использовании неокогнитрона [1];
2. Использовать некоторые инвариантные характеристики для распознавания, например, описанные в работе [2].

Первый подход имеет серьезные технические проблемы, второй - теоретические. Технические проблемы в первом подходе возникают из-за того, что практически невозможно заранее определить те вариации изображения объекта, которые будут необходимы во время реального распознавания объектов. Это также связано с трудоемкой задачей подбора изображений для обучения и их большим объемом, который нужно хранить в памяти нейронной сети. Второй подход имеет трудности, в первую очередь, в выявлении таких инвариантных характеристик, которые не изменялись бы при смещении объекта или изменении уровня освещенности. Кроме того, второй подход кажется неестественным, для распознавания используются не первоначально имеющиеся данные (собственно изображение объекта), а вторичные данные, например, при распознавании лиц используются расстояния между глазами человека, а также другие производные величины.

Данная статья предлагает третий подход, который, по крайней мере, в элементарных ситуациях способен решить проблему выделения и позиционирования объекта перед тем, как нейронная сеть займется распознаванием. Этот подход основан на следующей простой идее: если для нейронной сети затруднительно учитывать смещения объектов, то перед тем как помещать объекты в память нейронной сети, их нужно обработать детерминированными методами.

Эти методы в совокупности будем называть работой детектора движения. И тогда так называемый детектор движения должен будет выполнять следующие действия:

1. Отделять движущиеся объекты от фона;
2. Разделять одновременно движущиеся объекты;
3. Фильтровать слабое или локальное (хаотичное) движение, возникающее из-за внешних условий (ветер, снег, изменение освещенности).

## 2. Математическое обоснование невозможности применения параллельных алгоритмов в задачах предварительной обработки

В статье [3] автор обосновал следующие утверждения:

1. Проблем с распознаванием нелинейных входных данных у перцептрона нет;
2. Проблема возникает лишь тогда, когда входные данные оказываются инвариантными. Классическим примером является предикат четность;
3. ***Проблема инвариантности свойственна всем известным синхронным искусственным нейронным сетям;***
4. Для асинхронных нейронных сетей есть выход из этой проблемы, т.к. они помимо пространственных связей, рассматривают временные связи. Это дает определенную свободу, т.к. практически не могут существовать входные данные одновременно инвариантные как в пространстве, так и во времени. Но с асинхронными сетями связана другая проблема – устойчивости обучения, которой добиться намного сложнее;
5. Так как перцептрон является частным случаем параллельных вычислений, то проблема инвариантности свойственна и вообще параллельным вычислениям. Так в ходе анализа Минский показывает, что для ряда задач, таких как четность или связанность, следует отказаться от параллельных вычислений, т.к. они не оптимальны, а использовать последовательные вычисления. Таким образом, Минский как бы показывает границы применимости параллельных и последовательных вычислений. Примечательно, что причиной для таких исследований послужил перцептрон, что неоднократно подчеркивал Минский.

Далее рассмотрим ряд типовых задач предварительной обработки изображений во время видеонаблюдения.

**Типовая задача №1.** Если требуется построить машину для чтения печатных букв или цифр, то возникает естественное желание, чтобы эта машина могла распознавать их независимо от положения на странице, т.е. чтобы на решение машины не оказывали

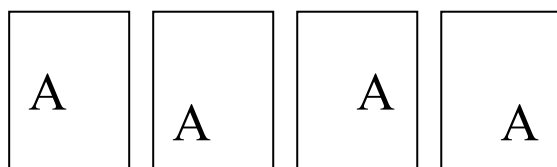


Рис.1. Различное положение букв на странице

влияния элементы группы переносов. Иными словами, восприятие машины должно быть “инвариантно относительно группы переносов”, т.е. ее решение должно быть одним и тем же для каждого представителя какого-либо класса эквивалентности относительно группы переносов.

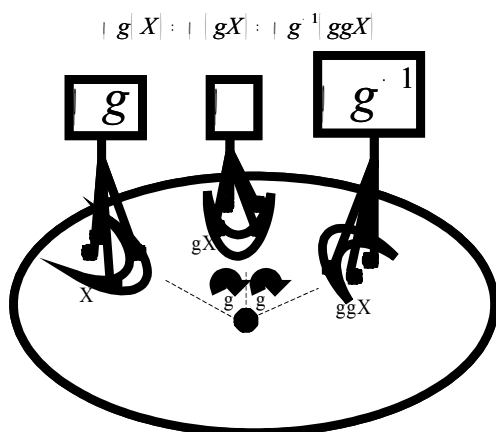


Рис. 2. Три предиката, эквивалентные относительно группы вращений

Понятие эквивалентности относительно группы перестановок (например, переносов, вращений) становится ясным из рис.2.

Итак, более точно первую задачу можно определить как – нахождение геометрического свойства. Вот как описывает Минский [4] понятие геометрического свойства: “Когда мы говорим *геометрическое свойство*, мы имеем в виду что-то инвариантное относительно переноса, или вращения, или растяжения. Сочетание первых инвариантностей определяет конгруэнтную

группу преобразований, а наличие всех трех дает фигуры *подобные* в евклидовой геометрии”.

**Типовая задача №2.** Ряд подзадач могут формулироваться различно, как (1) определить, одна или более фигур находятся в видимом пространстве, (2) плотная ли видимая фигура или же в ней находится отдельная фигура, (3) ... . Независимо от этого они имеют общую суть – нахождение признака, связана ли фигура, т.е. предиката связанность.

**Типовая задача №3.** Распознавание фигур в контексте, т.е., например, содержит ли данная фигура прямоугольник и, быть может, что-то еще.

Так вот, приведенные здесь три типа задач распознавания не рекомендуется решать с помощью параллельных способов вычислений, в том числе и с помощью перцептрона. Более того, проблема не в конкретной архитектуре перцептрона, а в необходимости знания всего глобального контекста. Поэтому как перцептрон, так и любые другие виды нейронных сетей неспособны полноценно решить представленные здесь задачи.

Мы не приводим здесь довольно сложные доказательства Минского [4, вторая часть], важно лишь, что они основываются на том, что перцептрон (равно как и любая другая нейронная сеть) не справляется с распознаванием инвариантных входных данных.

Альтернатива, предлагаемая Минским для данных задач, ясна и понятна – последовательные алгоритмы. Это еще раз показывает, что не все задачи могут быть решены параллельными методами, а роль эффективных последовательных методов довольно высока.

Но тем не менее автор решительно не согласен со следующим мнением, высказанным Минским. Он пишет:

“Мы убеждены, что вырождение способности перцептрона распознавать образы, если последние находятся в контексте других образов, служит серьезным предупреждением против использования его в реальных практических ситуациях. Конечно, этот недостаток можно смягчить, включая перцептрон в некоторый последовательный процесс (такой, в котором интересующая нас фигура на предварительной стадии изолируется и отделяется от своего контекста). Однако это предполагает достаточную способность к распознаванию на этапе предварительной обработки, чтобы можно было различить и устранить большинство обычно встречающихся контекстуальных помех. Иногда выполнить это гораздо труднее, чем провести собственно основную обработку”.

Именно таким способом, как описывает Минский, мы могли распознавать образы в контексте, включая перцептрон в последовательный процесс [5], или точнее, добавляя последовательный процесс предварительной обработки к основному процессу распознавания с помощью перцептрона. В данной же статье мы более точно опишем этот последовательный процесс предварительной обработки.

Центральной идеей, опровергающей предпосылки Минского, является то, что при движении объекта в контексте мы достаточно легко можем получить его контуры и содержимое. Поэтому процесс предварительной обработки элементарен, а перцептрон используется как распределенная память с повышением надежности, которая в ходе обучения настраивается на ассоциации с определенным сигналом управления.

Таким образом, система использует последовательную обработку как низкоуровневый процесс подготовки образов, на что сама нейронная сеть не способна. В то же время нейронная сеть перцептронного типа использовалась как высокоуровневое средство запоминания и классификации.

Надо сказать, что при реализации этой системы главным недостатком была низкая скорость, вследствие чего мы не смогли перейти от лабораторных испытаний к тестированию в реальном времени.

### 3. Необходимость исследования алгоритмов ИНС в других задачах

Итак, мы увидели теоретические недостатки перцептронов – проблема инвариантности и ее следствия. С другой стороны, есть еще одна проблема. Минский о ней пишет так:

“... мы покажем, что коэффициенты перцептрона могут расти значительно быстрее, чем показательная функция от  $|R|$ . Этот вывод имеет серьезные последствия, как практические, так и принципиальные: для запоминания коэффициентов нужна память большая, чем для записи всех изображений. Это ломает представление о том, что такая машина способна в некотором смысле абстрагировать”.

По поводу последнего можно дискутировать, так например, Минский строго показывает экспоненциальный рост коэффициентов лишь при инвариантных входных данных, которые в реальных примерах встречаются крайне редко.

Но, не вдаваясь в дальнейшую дискуссию, можно было бы сказать – стоп, а почему собственно перцептрон, разве еще не достаточно причин его не использовать, и где его вообще можно использовать?

Первое, идея о перцептроне бросает серьезный вызов математике. Казалось бы такая элементарная идея обучения с помощью подкрепления, и сетевидная архитектура. А какой приходится использовать арсенал математического аппарата, чтобы сделать попытки проанализировать этот механизм. В конце своей книги Минский отвечает на следующий вопрос:

“Для чего надо было доказывать все эти сложные теоремы? Разве нельзя было попросту взять перцептрон и посмотреть, может ли он распознать связность? Нет.”.

И он абсолютно прав. Кроме того, перцептрон поставил большое количество вопросов перед исследователями разных специальностей.

Второе, перцептрон представляет собой *распределенную память*, которая малоизученна и практически не используется в современном программировании. Именно из-за нее алгоритм работы перцептрона является не прозрачным и поэтому, несмотря на то, что Минский достаточно полно проанализировал перцептрон, он все же недопонимал некоторые принципы работы перцептрона, что подтверждают его слова:

“Несмотря на то что у нас нет так же хорошо разработанной теории обучения, мы можем, по крайней мере, показать, что в тех случаях, когда “обучение”, или “адаптация”, или “самоорганизация” все-таки происходит, это явление вполне объяснимо и не содержит ни малейшего намека на таинственные и малопонятные принципы поведения сложных систем. Действуют ли здесь такие принципы, мы не можем знать. Перцептрон никаких доказательств этому не дает, а наш успешный анализ перцептрона представляет еще одно косвенное подтверждение тезиса о том, что работоспособные кибернетические процессы понять можно, а те, которые нельзя понять, весьма подозрительны”.

И, в-третьих, многих исследователей не покидает чувство, что в перцептроне скрыты механизмы некоторой интеллектуальности. И это не случайно, так как первоначально перцептрон был разработан как модель работы некоторых особенностей высшей нервной деятельности. Современное состояние науки, кроме того, не может предложить альтернативной и столь же плодотворной модели.

В самом деле, что может предложить наука при решении задачи управления, кроме как наперед рассчитанных и полностью детерминированных действий – теории конечных автоматов. Все попытки найти причины интеллектуальности и возложить

некоторую техническую ответственность на другого: (1) развили теорию случайных процессов, (2) позволили построить теорию ограничений (которые накладываются на систему, действующую случайным образом).

Все это очень искусственно напоминает интеллект. Механизм же перцептрона более естественен, что позволяет говорить о нем как об интеллектуальном механизме.

Таким образом, перцептрон является в первую очередь лабораторной моделью, которая может позволить понять ряд глубоких принципов работы интеллектуальных систем. Именно поэтому, а не из практических соображений, стоит исследовать модель перцептрона.

#### **4. Описание эксперимента**

Говорить о предобработке данных в системе видеонаблюдения чисто теоретически невозможно, да и не имеет смысла, т.к. количество всех мыслимых случаев, которые пришлось бы рассмотреть (даже обобщая и классифицируя их на группы) близко к бесконечности. Поэтому во всех последующих рассуждениях мы будем опираться на конкретный эксперимент, описанный здесь.

Итак, видеокамера установлена таким образом, чтобы записывать определенный (статический) участок для наблюдения проезжей части. Длительность записи составляет 5 минут. Формат записи 24 кадра в секунду. Видеоданные поступают по каналу USB и записываются в формате MPEG с разрешением 320x240 пикселей, и далее разделяются на отдельные кадры. Каждый кадр приводится к формату графического файла PCX с 256 цветовой гаммой.

Для упрощения анализа используется лишь каждый 24-й кадр, т.е. сцена анализируется с точностью 1 кадр за секунду, что вполне достаточно при наблюдении. Таким образом, за время эксперимента должно быть проанализировано 300 кадров (5 минут по 60 секунд со скоростью 1 кадр в секунду).

На рис. 3. приведен первый кадр. Предполагается, что в нем нет движущихся объектов. Все присутствующие здесь объекты считаются статическими (на все время эксперимента), и поэтому такое изображение можно назвать фоном.

За время наблюдения было 5 движущихся объектов, которые должны были быть зафиксированы системой видеонаблюдения. Это следующие объекты:

1. Два человека (рис.4.), кадры №43 - №61, всего 19 кадров;
2. Велосипедист (рис.5.), кадры №117 - №123, всего 7 кадров;
3. Грузовик (рис.6.), кадры №130 - №136, всего 7 кадров;
4. Микроавтобус (разворачивается) (рис.7.), кадры №179 - №188, всего 9 кадров;
5. Микроавтобус (рис.8.), кадры №205 - №209, всего 5 кадров.



Рис. 3. Фон наблюдаемой области; все объекты статичны; движущихся объектов нет (Кадр №1)



Рис. 7. Движущийся объект «грузовик» (Кадр №132; объект присутствует на кадрах с №130 по №136)



Рис. 4. Движущийся объект «два человека» (Кадр №47; объект присутствует на кадрах с №43 по №61)



Рис. 5. Движущийся объект «велосипедист» (Кадр №118; объект присутствует на кадрах с №117 по №123)





Рис. 6. Движущийся объект «микроавтобус» (Кадр №182; объект присутствует на кадрах с №179 по №188)



Рис. 8. Движущийся объект «микроавтобус-2» (Кадр №207; объект присутствует на кадрах с №205 по №209)

## 5. Выделение движущегося объекта из фона

Основа любого детектора движения – это наличие функции выделения движущегося объекта из фона. От алгоритма данной функции зависят впоследствии все остальные вспомогательные функции, такие как фильтрование, разделение и масштабирование объектов. Вспомогательные функции настолько сильно связаны с функцией выделения, что говорить о них отдельно затруднительно. Тем не менее, в данном разделе описаны лишь основы данной функции, а ее расширения (нужные для других функций) будут описаны в соответствующих разделах.

Процесс анализа кадров разделен на этапы. В каждом этапе анализируется 4 последовательных кадра в два прохода. А именно, первый кадр из имеющихся четырех последовательно сверяется со вторым, третьим и четвертым. В результате получается три кадра, которые во втором проходе опять сверяются первый со вторым, и полученный результат с третьим.

Таким образом, каждый этап позволяет анализировать кадры, исключая те движения, которые не являются основными (например, блики, движение листьев от ветра и т.п.).

Здесь мы рассмотрим алгоритм сверки двух произвольных, но последовательных кадров. Основным настраиваемым параметром являются две величины, от которых впоследствии зависит анализ минимально существенного движения:

1. Минимальная область движения  $Z$  ( $10 \times 10$ )

Определяет величину области, внутри которой движение не анализируется - в нашем эксперименте она принята  $10 \times 10$  пикселей.

2. Количество изменений, которые нужно считать существенными  $K_z$  (определяется для области  $Z$ )

Определяет то количество отличий в области  $Z$  на двух разных кадрах, наличие которых позволяет сказать, что области разные. В нашем примере 10 пикселей (или 10% от всей области  $Z$ ).

Тогда алгоритм сводится к набору элементарных действий:

1. Первый и второй кадр разделяем на одинаковые области в соответствии с размерами области  $Z$ ;
2. По очереди сверяем все полученные области;
3. Если точки в двух соответствующих областях на разных кадрах совпадают, то увеличиваем счетчик одинаковых точек  $P_z$ ;

4. Если  $P_z > K_z$ , то удаляем с изображения соответствующую область, т.е. если область изменилась незначительно (количество одинаковых точек больше уровня существенного движения), то считаем, что в данной области движения не было, и эта область дальше не анализируется;
- 4'. Для второго прохода в п.4. меняется логика – области, где было движение, уже выделены, поэтому нужно убрать те области, где области не совпадают на последовательных кадрах. Формально это выполняется тем же алгоритмом, только меняется условие на  $P_z < K_z$ , то есть удаляем с изображения те области, где количество совпадающих точек меньше необходимого минимума совпадений (уровня существенного движения).

## 6. Разделение одновременно движущихся объектов

При наблюдении существуют ситуации, когда в одном кадре могут появиться совершенно разные движущиеся объекты. В нашем примере такой ситуации нет. Но рассмотрение этой ситуации важно, т.к. разделение одновременно движущихся объектов, помогает решить задачу фильтрации слабых движений.

Разными будем считать такие объекты, которые при движении не пересекаются и движутся в разных областях наблюдаемой сцены. Если воспользоваться результатом функции «выделения движущегося объекта из фона», то остается проанализировать, связаны ли те области, которые остались на изображении. Связанными мы будем считать те области, которые находятся рядом с рассматриваемой областью с любой из восьми сторон. Если области полностью не связаны, то мы имеем дело с двумя разными объектами, иначе, если есть хотя бы одна связка, то это один и тот же объект.

Разделяя таким образом объекты, мы можем в дальнейшем работать с ними отдельно и последовательно. Алгоритм разделения сводится к следующим шагам:

1. Берем первую область и смотрим, с чем она связана – добавим координаты связанных областей в стек;
2. Вычеркиваем рассмотренную область;
3. Берем область с координатами из стека и делаем операцию п.1.;
4. Повторяем п. 1-3 для всех областей, занесенных в стек;
5. Если общее количество областей равно прошедшему через стек количеству областей, то объект один; иначе - повторить весь алгоритм для оставшихся областей.

## 7. Фильтрация слабых и локальных движений

Функция фильтрации заложена в уже рассмотренные алгоритмы выделения и разделения. Когда мы говорим о слабом движении, например, бликах от изменения освещенности, мы предполагаем, что реального движения нет, и такое псевдодвижение нет необходимости контролировать. Поэтому на такое движение детектор движения не должен реагировать. Когда мы говорили о параметре «уровень существенного движения» в функции выделения движущегося объекта из фона, то игнорирование слабого движения (псевдодвижения) уже было учтено.

Дополнительно к этому имеет смысл отдельно рассматривать локальные (хаотичные) движения, например, движение листьев под воздействием ветра. Такое движение, в отличие от слабого движения, может быть сильным, но лишь в отдельных областях. Разделяя кадр на отдельные объекты с помощью описанной выше функции, видим, что существуют области не связанные ни с какой другой областью. Если предположить, что реальный объект состоит как минимум из двух связанных областей, то единичные области можно считать областями локального (хаотичного) движения и в дальнейшем их можно не учитывать.



## 8. Тестирование предложенных алгоритмов

Результаты тестирования представлены в таблице 1. Из нее видно, что алгоритм «выделение движущегося объекта из фона» отобрал 76 кадров из 300, где есть хоть малейшее движение. Изменение освещенности и движение листьев дерева слева сверху на кадре и листьев куста справа снизу вызывают слабые и локальные (хаосные) движения. Анализ на связанность выделенных областей алгоритмом «разделение одновременно движущихся объектов» позволяет исключить из рассмотрения 43 кадра из ранее отобранных 76. Таким образом, в 33 кадрах присутствует движущийся объект. Причем алгоритм ошибается лишь в 3-х, где на самом деле нет существенного движения.

Таблица 1

### Результат работы детектора движения

Функция выделения				Функция разделения			
I	II	III	IV	V	VI	VII	VIII
Нет дв.	1	17	2	-			
Объект №1	2	43	1	-			
	3	44	2	2	3-3	7-8	1x2
	4	45	2	2	5-5	7-8	1x2
	5	46	1	-			
	6	47	2	2	8-8	8-9	1x2
	7	48	1	-			
	8	50	1	-			
	9	51	1	-			
	10	54	1	-			
	11	55	2	2	21-21	8-9	1x2
Нет дв.	12	65	1	-			
Объект №2	13	117	3	3	25-27	18-19	3x2
	14	118	5	5	20-21	14-17	2x4
	15	119	3	3	14-16	15-15	3x1
	16	120	3	2	11-11	13-14	1x2
	17	122	2	2	5-5	10-11	1x2
Нет движения	18	126	1	-			
	19	127	1	-			
	20	128	3	2	31-31	17-18	1x2
	21	129	4	2	29-30	16-17	2x2
Объект №3	22	130	38	37	27-32	13-24	6x12
	23	131	157	146	10-32	9-22	23x14
	24	132	104	103	6-22	8-18	17x11
	25	133	66	64	6-16	8-16	11x9
	26	134	60	60	4-13	7-14	10x8
	27	135	43	41	2-8	6-12	7x7
	28	136	17	17	1-3	6-11	3x6
	29	138	1	-			
Нет движения	30	144	1	-			
	31	156	1	-			
	32	170	1	-			
	33	177	1	-			
	34	178	2	-			
	35	179	3	2	31-32	4-4	2x1
Объект №4	36	180	3	3	30-32	4-4	3x1
	37	181	7	7	29-32	3-4	4x2
	38	182	6	6	28-31	3-4	4x2
	39	183	4	4	28-30	3-4	3x2
	40	184	3	3	29-30	3-5	2x3
	41	185	2	2	30-30	4-5	1x2
	42	186	2	2	30-31	5-5	2x1
	43	187	4	4	30-32	4-5	2x2

Функция выделения				Функция разделения			
I	II	III	IV	V	VI	VII	VIII
Нет дв.	44	191	1	-			
Объект №5	45	192	1	-			
	46	205	4	4	30-32	8-9	3x2
	47	206	16	16	22-30	7-9	9x3
	48	207	17	17	16-23	7-10	8x4
	49	208	17	16	8-14	7-10	7x4
	50	209	17	17	1-7	6-9	7x4
	51	210	1	-			
	52	211	1	-			
	53	221	1	-			
	54	225	1	-			
Нет движения	55	228	1	-			
	56	231	1	-			
	57	232	1	-			
	58	233	1	-			
	59	236	1	-			
	60	253	1	-			
	61	254	1	-			
	62	255	1	-			
	63	256	1	-			
	64	268	1	-			
	65	269	1	-			
	66	271	1	-			
	67	273	1	-			
	68	275	1	-			
	69	277	1	-			
	70	278	1	-			
	71	279	2	-			
	72	281	2	2	30-30	10-11	1x2
	73	284	1	-			
	74	285	2	-			
	75	289	1	-			
	76	293	2	-			

I – Какой объект движется в это время; II – № п/п; III – кадры, на которых зафиксировано движение; IV – Количество областей, в которых зафиксировано существенное движение; V – Количество связанных областей; VI – Расположение области по оси x; VII – Расположение области по оси y; VIII – Размер области.

И наоборот, там, где было реальное движение объектов, алгоритм не зафиксировал следующее количество кадров:

1. при движении первого объекта – из 19 кадров не зафиксировано 15;
2. при движении второго объекта – из 7 кадров не зафиксировано 2;
3. при движении третьего объекта – из 7 кадров не зафиксировано 0;
4. при движении четвертого объекта – из 9 кадров не зафиксировано 0;
5. при движении пятого объекта – из 5 кадров не зафиксировано 0.

## 9. Заключение

Объясняя полученные результаты нужно учитывать площадь объекта, занимаемую на изображении и его скорость. Тогда становится понятно почему для первых двух объектов на

некоторых кадрах не зафиксировано их движение – эти объекты маленькие и их движение в ряде случаев воспринимается алгоритмом как слабое движение.

При анализе движения велосипедиста единственной проблемой данного объекта, является его малая величина, из-за чего его трудно зафиксировать и в последствии правильно идентифицировать. Из-за того, что объект маленький, но быстро движется, его движение приближается к хаотичному, хоть и не является таким.

При анализе движения грузовика есть тоже свои особенности. Связаны они в первую очередь с медленным движением большого объекта. Это приводит к тому, что смешиваются области движения с областями, где движение будет только в последующих кадрах. Избежать этой проблемы можно если анализировать движение с другой частотой (например, через каждые 2 сек., а не каждую секунду).

Наилучшие результаты получены при движении микроавтобуса. Это связано в первую очередь с тем, что величина объекта была пропорциональна его скорости. В результате мы получаем достаточно чистое изображение, которое можно использовать в дальнейшем анализе с использованием искусственных нейронных сетей.

Наличие предложенных функций у детектора движения, выполняющего все действия по предобработке, позволяет помещать в память нейронной сети уже более пригодные для сравнения и последующего анализа объекты, а также память нейронной сети не перегружена ненужной информацией, которую в противном случае пришлось бы анализировать средствами нейронной сети, которые для этого практически не предназначены. Кроме того, все это отражается на скорости работы, ведь достаточно простые детерминированные алгоритмы требуют меньше ресурсов памяти и времени по сравнению с нейронной сетью. А именно здесь, на этапе предварительной обработки, вся мощь нейронной сети не нужна, и поэтому ее следует заменить более простыми и быстрыми детерминированными алгоритмами.

## References

1. Вассерман Ф. (1992). *Нейрокомпьютерная техника: теория и практика*, Мир, Москва.
2. Самаль Д.И. (1998). Построение систем идентификации личности на основе антропометрических точек лица, Цифровая обработка изображений, Минск, ИТК НАНБ, с. 72-78.
3. Яковлев С. (2006). *Линейность и инвариантность в искусственных нейронных сетях*, <http://evrika.tsi.lv/index.php?name=texts&file=show&f=409>
4. Минский М. (1971). *Перцептроны*, Мир, Москва.
5. Яковлев С. (2004). *Система распознавания движущихся объектов на базе искусственных нейронных сетей*. ИТК НАНБ, Минск, ?с.

Sergey Jakovlev, Mg.sc.eng., Institute of Information Technology, Riga Technical University, 1 Mezha Street, E-mail: sergeyk@fis.lv, tac@inbox.lv

*Jakovlevs Sergejs.*

*Jakovlev Sergey.*

***Яковлев Сергей. Предварительная обработка в системе видеонаблюдения на основе искусственных нейронных сетей***

**Аннотация.** В статье показана необходимость предварительной обработки последовательными алгоритмами, которые должны устранить ограничения, накладываемые искусственными нейронными сетями, при решении задачи распознавания изображений. Показаны ограничения параллельных алгоритмов, в том числе и искусственных нейронных сетей, при решении задач выделения движущегося объекта из фона, разделения одновременно движущихся объектов и т.п. Предлагается при решении таких задач применять исключительно последовательные алгоритмы с целью четко позиционировать объект перед распознаванием с помощью искусственной нейронной сети. Проведены эксперименты применения таких последовательных алгоритмов и показаны их возможности.