

***Идентификация спортивных игроков во время
телевизионной трансляции матча
Национальной хоккейной лиги***

*Работу выполнил студент
3 курса 5 группы ММФ БГУ
Кирилло Дмитрий*

Минск, 2021

Оглавление

Бизнес-задача	2
Формализация задачи в терминах Machine Learning	4
Описание данных для решения задачи	4
Метрики качества и схема валидации	8
Описание базового non Machine Learning решения	9
Варианты решения с помощью Machine Learning	11

Бизнес-задача

В данной работе рассмотрим задачу идентификации спортивных игроков(хоккеистов) во время трансляции матча Национальной хоккейной лиги (в дальнейшем НХЛ).

Идентификация игроков и арбитров во время матча является сложной задачей компьютерного зрения из-за схожести внешнего вида игроков, размытости черт лица, возникшими перед спортсменами помех в виде различных препятствий и физических особенностей. Однако, благодаря особенности видеотрансляций ТВ компаний, мы можем наблюдать за игроками и арбитрами во время нахождения их непосредственно на игровом поле, обрабатывая последовательности изображений игроков разной длины.

Решение данной задачи является подход, основанный на глубокой рекуррентной сверточной нейронной сети. Поскольку игровой номер на джерси является наиболее отличительной и идентификационной чертой хоккеиста, то проблема решается через остаточную нейронную сеть (ResNet) совместно с сетью долгой краткосрочной памяти (LSTM) для обнаружения и идентификации особенностей игровых номеров на джерси хоккеиста и позволяющую информации нейронной сети из предыдущих кадров влиять на прогнозы передвижений игроков для будущих временных интервалов.

Важное замечание: наша задача не рассчитана на анализировании или сборе статистических данных игроков и(или) арбитров! Только высвечивании нужной ТВ компании информации о хоккеисте или арбитра во время их нахождения на игровой площадке.

Прежде чем распознавать игровой номер хоккеиста, мы должны взять базу данных игроков НХЛ. Стоит отметить, что каждая спортивная ТВ компания имеет разновидности данных на игроков и арбитров, то есть одни рассматривают определенную статистику, взятые из различных статистических сайтов либо же

поддерживаемыми самими ТВ компаниями, другие берут данные непосредственно с сайта НХЛ. Однако, наша задача состоит в том, что все нужные нам данные имеются у всех баз данных и датасетов: игровой номер хоккеиста, его фамилия и имя, а также принадлежность игрока к команде. Если же мы рассматриваем случай арбитра, то у него имеются все такие же данные, что у игроков, за исключением того, что арбитр не принадлежит ни к какому классу клубов, а к классу арбитры НХЛ. Все дальнейшие разновидности данных для игроков или арбитров нас не интересуют, так как ТВ компании сами будут высвечивать нужную им информацию во время матча.

Примерная структура общей базы данных выглядит следующим образом:

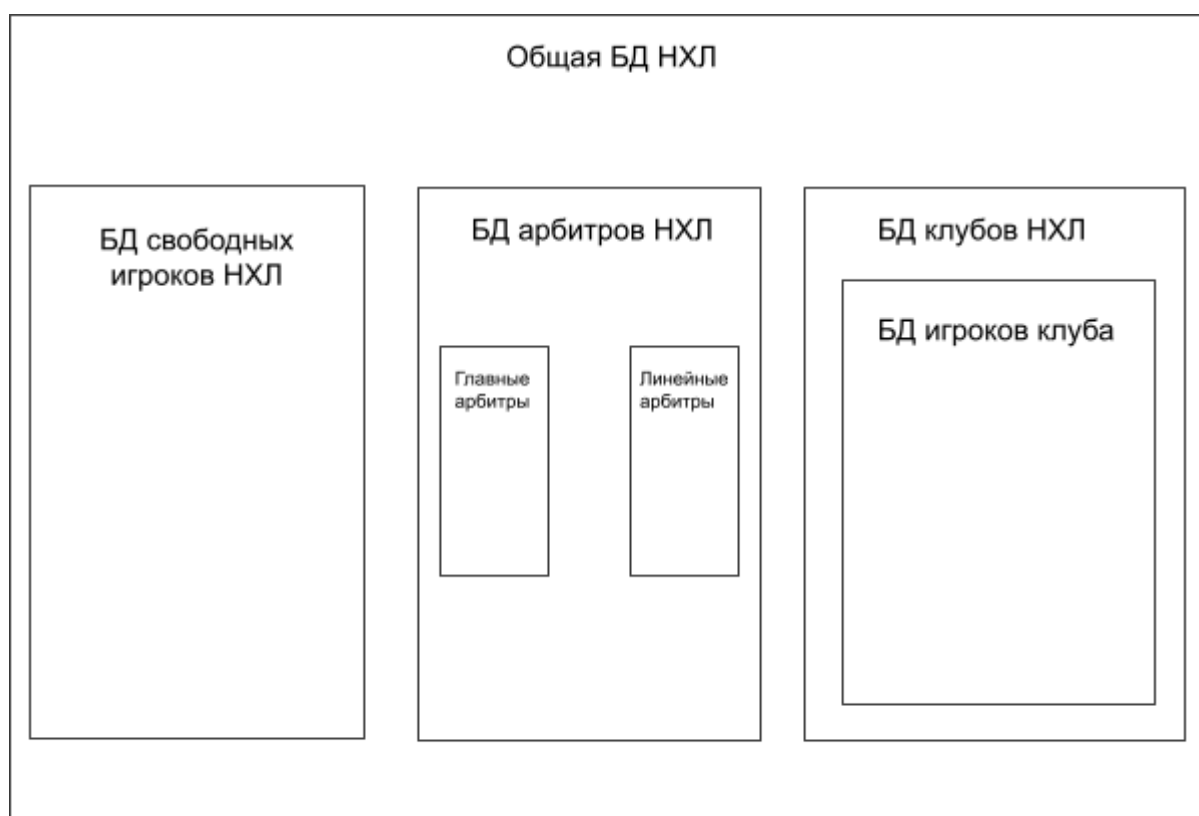


Рисунок 1. Структурная схема общей базы данных НХЛ

Как можно заметить, данная база данных будет периодически обновляться, но при этом сама структура никак не будет изменяться.

Отметим также, что информация о игроке скрывается, когда хоккеист пропадает из “поля” зрения во время телетрансляции.

Формализация задачи в терминах Machine Learning

Для формализации задачи сначала необходимо разобраться, какие данные используются на входе и какие данные ожидаются на выходе. В данной задаче

идентификации игроков на вход алгоритма подается видеофрагмент (или определенная последовательность кадров) игрока на игровой площадке, и в итоге возвращается его игровой номер на джерси. Например, на рисунке ниже показана последовательность кадров движущегося игрока (также называемая треклетом). На выходе алгоритма должно получиться число, т.е. номер на его джерси.



Рисунок 2. Пример треклета с разборчивым и неразборчивым номером

У данной бизнес-задачи есть несколько путей решения, использующих различные методы машинного обучения. Однако, наиболее подходящим методом решения является классификация видеофрагментов. Она заключается в нахождении глобальных дескрипторов видео, которые включают в себя не только пространственную информацию (где на кадре находится объект), но и динамическую временную информацию, получаемую от движения, происходящего на протяжении нескольких последовательных кадров, объекта. В данном случае, необходимо закодировать треклеты игроков, обращая внимание на расположение номера джерси, а также на изменение его внешнего вида при движении игрока. Таким образом удастся наблюдать за игроком в течение какого-то промежутка времени, нежели чем на отдельных кадрах, на которых номер может быть размыт или вообще не виден.

Описание данных для решения задачи

В качестве одних из данных, которые мы будем использовать при обучении сверточной нейронной сети, возьмем из GitHub: <https://github.com/grant81/hockeyTrackingDataset>. Базу данных игроков, а также арбитров, можно взять, к примеру, из сайта <http://www.nhl.com/>.

Важно отметить, что игровой номер хоккеиста варьируется в промежутке от 1 до 98, так как 99 номер выведен из обращения Лигой в честь легендарного игрока Уэйна Гретцки за его выдающиеся заслуги в НХЛ. При этом, в одной команде не может быть два повторяющихся игровых номера, но в разных клубах такая вариация может существовать. Также стоит сказать, что клубный идентификационный ключ состоит из уникальной комбинации трех латинских букв.

Перейдем к рассматриваемым нами датасетам. Из базы данных игроков и арбитров нам потребуются следующие переменные:

- PlayerNumber - игровой номер хоккеиста. По этой переменной мы будем сопоставлять нашу сверточную нейронную сеть для идентификации игрока;
- ClubID - идентификационный номер хоккейного клуба, за который выступает игрок. Состоит из уникальной комбинации трех латинских букв;
- Name - имя хоккеиста;
- Surname - фамилия хоккеиста;
- Position - позиция игрока (вратарь/защитник/нападающий).

Данные из сайта GitHub: <https://github.com/grant81/hockeyTrackingDataset> выглядят следующим образом:

- frame_number - порядковый номер кадра в видеофрагменте;
- Player_Id - идентификационный номер рассматриваемого хоккеиста;
- left_corner_x - отметка рассматриваемого игрока в bounding box по оси X, отсчитываемая слева;
- left_corner_y - отметка рассматриваемого игрока в bounding box по оси Y, отсчитываемая слева;
- height - высота bounding box, отсчитывается с переменной left_corner_y;
- width - ширина bounding box, отсчитывается с переменной left_corner_x;
- detection_confidence - коэффициент распознавания игрового номера на джерси хоккеиста. Значение варьируется от 0 до 1.
- tracklet_id - последовательность изображений игроков переменной длины за кадр;
- visibility - видимость изображения хоккеиста на площадке. Значения данной переменной либо 0.5, либо 1.0.

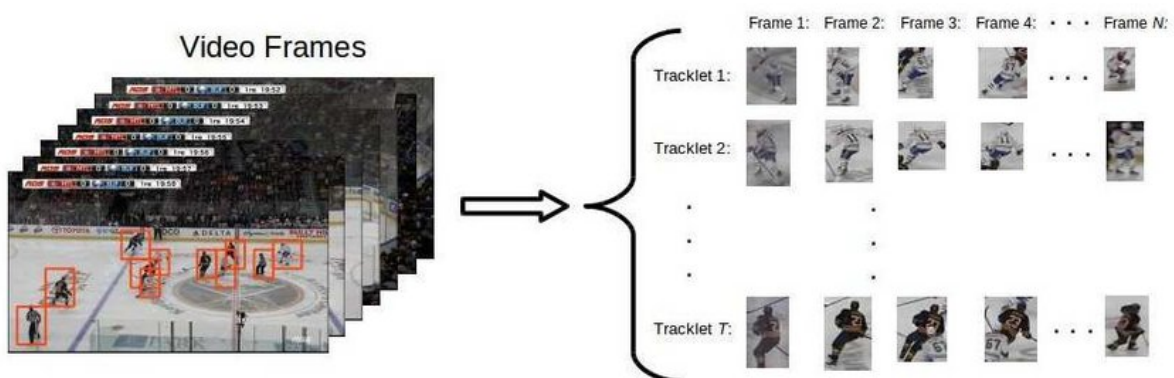


Рисунок 2. Принцип взаимодействия переменных *frame_number* и *tracklet_id*

Стоит отметить важное замечание к переменной *Player_Id* из датасета GitHub: данная переменная рассматривается как изучение некоего объекта под названием хоккеист. Данный объект изучается нейронной сетью до тех пор, пока телевизионная камера находится в “поле” зрения ТВ картинки. В случае, если игрок находится на игровой площадке, но вышел за пределы телекартинки и вновь въехал в зрительское обозрение, то нейронная сеть посчитает, что рассматривается новый хоккеист и начнет вычисление игрового номера на джерси спортсмена по-новому. Соответственно, проводятся и новые вычисления порядкового номера кадра в видеофрагменте и других переменных.

Переменная visibility имеет два значения: 0.5 или 1.0. Это связано с тем, что если игрок находится рядом с помехами в распознавании или закрыт какими-либо объектами, то кадр принимает значение 0.5, то есть плохо различает игровой номер хоккеиста. Значение 1.0, соответственно, показывает, что игрок находится без каких-либо препятствий и его легко можно сегментировать. Также, по словам создателя датасета, степень достоверности обнаружения всегда равна единице, потому что набор данных был аннулирован вручную.

Теперь рассмотрим наш набор данных непосредственно перед хоккейным матчем. Как известно, хоккейный матч состоит из двух команд (одновременно на игровой площадке от одной команды максимально может находиться 1 вратарь + 5 полевых игроков, минимально - 1 вратарь и 3 полевых игрока) и четырех арбитров (2 главных и 2 линейных арбитра), каждая из которых на определенный матч имеет свои цвета игровых форм. Каждая из команд минимум за час до начала матча подает свой список из игроков, которые будут участвовать в матче, а Лига, в свою очередь, назначает арбитров за несколько дней. То есть, перед каждым матчем формируется протокол состава команд и арбитров. Это поможет нам ограничить выбор соответствия игрового номера хоккеиста и увеличить точность распознавания номера игрока. Тогда схему работы нашей задачи можно определить следующим образом:

Подготовка данных перед телетрансляцией:

- выбираем из нашей базы данных два клуба;
- выбираем, согласно протоколу матча, двух главных и двух линейных арбитров ;
- выбираем, согласно протоколу матча, для каждого из клубов список участвующих в матче хоккеистов;
- определяем цвета формы для каждого из клубов (цвет формы арбитров не изменяется по ходу сезона).

Тогда, во время телетрансляции, мы должны проделать следующие шаги для решения нашей задачи:

- определить, при помощи нейронных сетей, цвет игровой формы и номер хоккеиста на джерси;
- сопоставить найденный номер с цветом клуба и находим из списка хоккеистов нужного нам игрока;
- высветить нужную информацию на хоккеиста во время телеэфира.



Рисунок 3. Пример работы обнаружения спортсменов на площадке

Наглядная ниже схема показывает, как работает алгоритм высвечивания информации над хоккеистом во время его пребывания на игровой площадке:

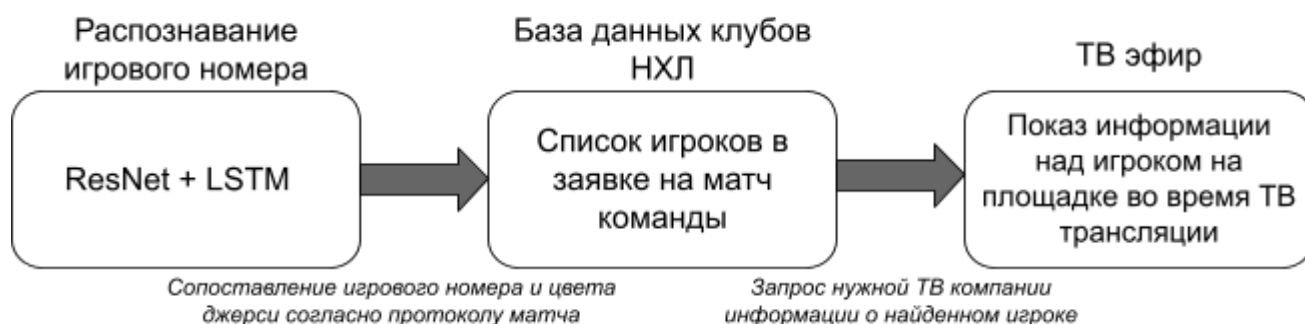


Рисунок 4. Структурная схема алгоритма высвечивания информации над хоккеистом во время его пребывания на игровой площадке

В итоге, после проделанной работой алгоритма, мы должны получить нижеприведенное телекартинку в эфире телеканала:



Рисунок 5. Итоговый полученный кадр во время телевизионной трансляции с информацией над игроками

Как уже писалось ранее, информация о игроке скрывается, когда хоккеист пропадает из “поля” зрения во время телетрансляции.

Метрики качества и схема валидации

Используемые далее метрики используют следующие типы исходов: истинно положительные (TP), истинно отрицательные (TN), ложно положительные (FP) и ложно отрицательные (FN). Для наглядности их можно представить в следующей таблице:

		Предсказанные	
		Отрицательные	Положительные
Настоящие	Отрицательные	TN (True Negative)	FP (False Positive)
	Положительные	FN (False Negative)	TP (True Positive)

Рисунок 6. Таблица типов исходов

С помощью этих типов можем дать определение используемым метрикам:

1) Ассурасу (Точность)

Показывает количество правильно проставленных меток класса (истинно положительных и истинно отрицательных) от общего количества данных:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision (Точность)

Эта точность показывает количество истинно положительных исходов из всего набора положительных меток:

$$Precision = \frac{TP}{TP + FP}$$

3) Recall (Полнота)

Определяет количество истинно положительных среди всех меток класса, которые были определены как «положительный»:

$$Recall = \frac{TP}{TP + FN}$$

4) F1-score

Является функцией от Precision и Recall:

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall}$$

При решении данной задачи предлагается использовать все из вышеперечисленных метрик для построения наилучшей модели. Однако, т.к. ложные предсказания не критичны, то достаточно использовать только Accuracy.

Важно отметить, что одной из проблем нашей задачи состоит в том, что, во время обнаружения игрока и высвечивании информации над ним, наша нейронная сеть должна уметь убирать информацию о игроке, а точнее bounding box, когда хоккеист приехал на скамейку запасных, и определять новый bounding box, когда хоккеист уехал из скамейки запасных. Существует несколько решений данной проблемы:

- довериться нейронной сети, которая точно вычисляет в процессе обучения местоположения игроков, а также оттенки цветов, препятствий и т.д. для хоккеиста, что приведет к распознаванию тех мест, где bounding box должен либо убратся, либо появиться;
- провести дополнительную разметку игровых бортов в тех местах, где игрок может заехать на скамейку запасных. Данная задача не является особо затратной и долгой, так как, из-за положения телевизионных камер, мы изначально определяем наши “особые” места слежения и присваивания/устранения объекту bounding box для каждой определенной камеры. Стоит отметить, что в этом нам также помогает технология размещения рекламы на бортах при помощи нейронной сети. Если такая технология уже присутствует у ТВ компании, то нам достаточно указать регион возможной смены игрока.

Описание базового non Machine Learning решения

Одним из базовых решений высвечивание информации над игроком во время ТВ трансляции не машинного обучения является метод подсветки информации игрока(-ов) напрямую при помощи человека, а именно редактора трансляции, не требующего какого-либо познания в машинном обучении. Однако человек, высвечивающий трансляцию во время прямого эфира не в моменте повтора, имеет несколько проблем:

- невозможность предугадывания действий хоккеиста и его направления движения;
- относительная трудность подсвечивать информации сразу для нескольких игроков. Однако, если редакторов трансляции большое количество, то эта проблема легко решается за счет количества “рабочих” рук;
- несвоевременное удаление информации о игроке во время его не “попадания” в кадр или приезда на скамейку запасных

Небольшой плюс решения данным методом является то, что человек может распознать хоккеиста даже с учетом каких-либо помех в кадре, а также засчет некоторых особенностей игрока таких как катания, шнурков, передач, бросков и т.д. Также человек может распознать игровой номер, который находится на рукавах хоккеиста, в то время как нейронная сеть может не развидеть этого из-за маленького масштаба цифр на джерси спортсмена.

Стоит отметить, что существуют также специальные программы, где легко производить разметку игроков, в следствии которой над игроком высвечивается нужная информация во время эфира. К примеру, одной из таких программ является приложение от SPORTLOGiQ, как показано это на рисунке ниже. Данная программа обладает большим спектром для улучшения графики разметки, а также улучшения ТВ трансляции во время матча НХЛ. Стоит отметить, что данная компания заключило сотрудничество непосредственно с самой НХЛ.

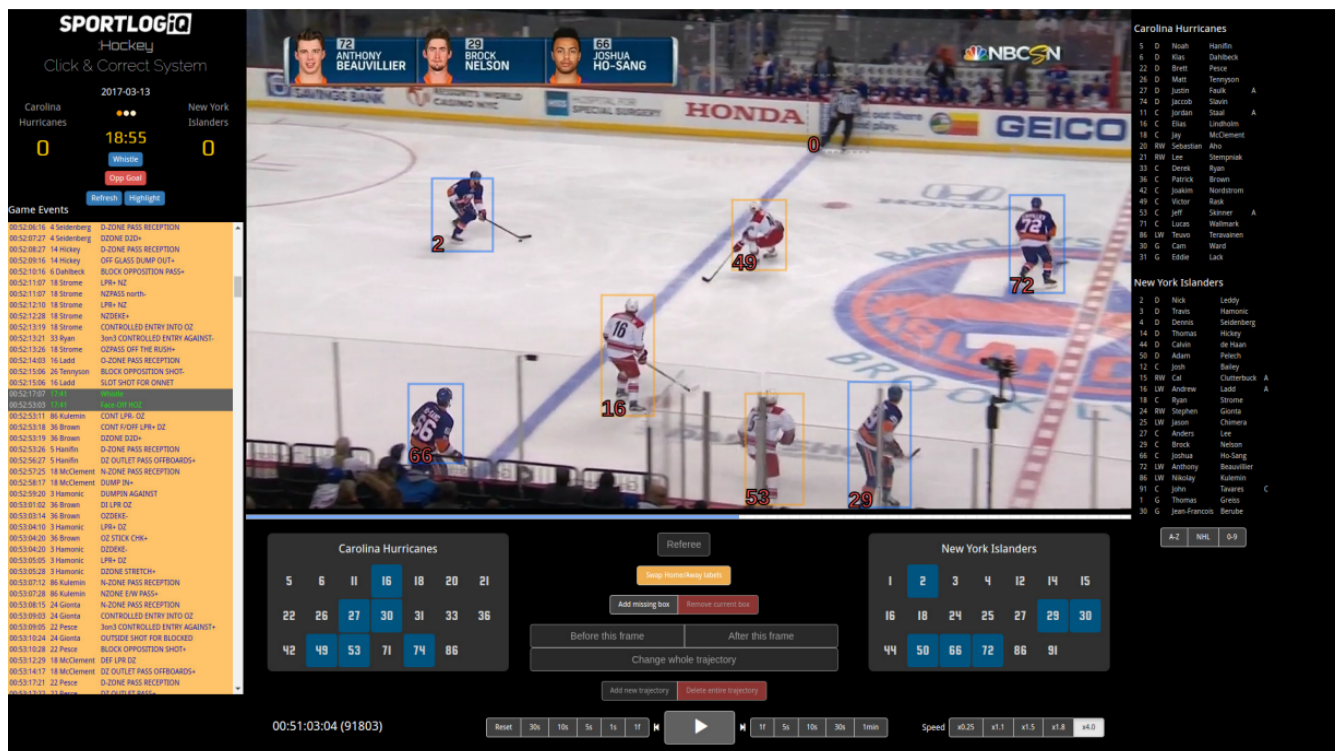


Рисунок 7. Пример программы решения показа информации не машинного обучения

В остальных случаях, использование каких-либо решений без использования машинного обучения в рамках этой задачи считается не логичным и затратным, с точки зрения финансов, так как содержание как минимум одного редактора, который бы размечал данные и(или) выводил бы потом вручную информацию во время

телевизионной трансляции, обходится гораздо дороже, чем использование уже настроенной и обученной нейронной сети для поставленной нами задачи.

Варианты решения с помощью Machine Learning

Сверточные нейронные сети отлично справляются в задачах распознавания элементов на изображении, благодаря изучению сложных и комплексных признаков, однако они могут только работать с пространственными данными статических изображений. В задаче классификации видеофрагментов, с другой стороны, также доступна дополнительная информация из-за динамического свойства видео. Для того, чтобы воспользоваться этой информацией, необходимо построить модель, умеющую работать с последовательностью данных - в нашем случае с треклитами. Таким образом, изучая и используя эту динамическую информацию, можно прийти к более точным предсказаниям.

Поэтому для решения данной задачи предлагается модель нейронной сети, состоящая из остаточной сверточной нейронной сети (Residual network, или ResNet) вместе с рекуррентными слоями долговременной краткосрочной памяти (Long short-term memory, или LSTM). Сверточная нейронная сеть отвечает за изучение признаков номера джерси, а рекуррентный слой позволяет информации передаваться от кадра к кадру. Вместо обычной сверточной нейронной сети предлагается использовать остаточную, чтобы оптимизировать процесс обучения и минимизировать ошибку, получаемую во время тренинга сети из-за большого количества слоев.

Данная модель берет на вход треклет игрока и для каждого кадра возвращает вектор классовых вероятностей. Затем, находится среднее между всеми векторами, чтобы получить финальный вектор классовых вероятностей для данного треклета. За номер джерси берется максимальная вероятность из полученного вектора. Общий алгоритм сети показан на рисунке ниже.

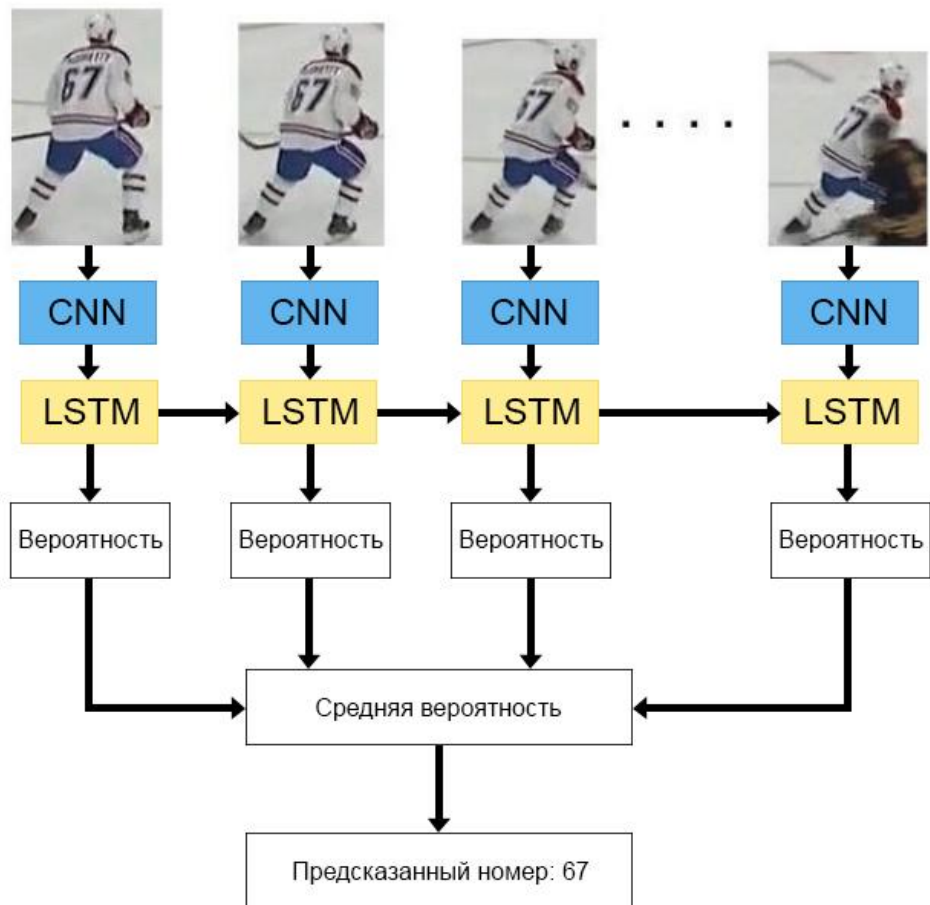


Рисунок 8. Алгоритм предложенной модели нейронной сети