ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 109 за 2021 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

А.В. Подопросветов, М.Е. Лузина, А.П. Алисейчик, Е.В. Павловский, И.А. Орлов

Применение нейронных сетей для предсказания физических действий человека

Рекомендуемая форма библиографической ссылки: Применение нейронных сетей для предсказания физических действий человека / А.В. Подопросветов [и др.] // Препринты ИПМ им. М.В.Келдыша. 2021. № 109. 16 с. https://doi.org/10.20948/prepr-2021-109
https://library.keldysh.ru/preprint.asp?id=2021-109

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

А.В. Подопросветов, М.Е. Лузина, А.П. Алисейчик,
Е.В. Павловский, И.А. Орлов

Применение нейронных сетей для предсказания физических действий человека

Москва — 2021

А.В. Подопросветов, М.Е. Лузина, А.П. Алисейчик, Е.В. Павловский, И.А. Орлов

Применение нейронных сетей для предсказания физических действий человека

Работа посвящена предсказанию физической активности человека. В статье предлагается алгоритм машинного обучения для прогнозирования физических действий человека, созданный с использованием программного пакета для сбора данных, обработки данных и обучения алгоритма регрессии на обработанных данных. Полученные результаты связаны с автоматическим определением начала и вида физического действия, совершаемого человеком. Работа направлена на совершенствование систем управления для промышленного применения экзоскелетов, предназначенных для увеличения силы человека за счет внешнего каркаса. В будущем возможно использование результатов исследования для более совершенного взаимодействия со вспомогательными устройствами на предприятиях.

Ключевые слова: предсказание физической активности, нейронные сети, обработка данных, промышленные экзоскелеты, прогнозирование действий.

Podoprosvetov Aleksei Valerievich, Luzina Margarita Evgenievna, Pavlovsky Evgeny, Aliseychik Anton Pavlovich, Orlov Igor Aleksandrovich

Neural networks for predicting human actions

This work is devoted to the prediction of human actions. The article proposes a machine learning algorithm for predicting human physical actions, created using a software package for collecting data, processing data and training a regression algorithm on the processed data. The results obtained are associated with the automatic determination of the beginning and type of physical action performed by a person. The work is aimed at improving control systems for industrial use of exoskeletons designed to increase human strength through an external frame. In the future, it is possible to use the research results for better interaction with assistive devices in enterprises.

Key words: action prediction, neural networks.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 20-31-90101.

Введение

Развитие совокупности методов машинного обучения, называемых глубоким обучением, позволило создавать сложные архитектуры нейросетей, которые обладают достаточной производительностью и дают возможность решать широкий спектр задач в области компьютерного зрения, не поддававшихся эффективному решению ранее. В частности, становится остро актуальной для исследования в наше время задача распознавания и предсказания действий человека с помощью методов машинного обучения. С развитием технологий растет и число возможных применений: взаимодействие человека с технологическим оборудованием и роботами, анализ производительности и контроль безопасности рабочих на предприятии, системы распознавания для видеонаблюдения, формирование реалистичных движений в кино и компьютерных играх и пр. Принципиальная особенность технологического прогресса состоит в необходимости изобретать новые способы обработки и анализа данных, а также методы взаимодействия с ними и адаптации под современные нужды. Данная работа направлена на улучшение систем управления для применения активных или полуактивных промышленных экзоскелетов - устройств, предназначенных для увеличения силы человека за счёт внешнего каркаса.

Целью данной работы является создание нейронной сети, которая сможет прогнозировать тип физической активности человека и момент её начала на основе существующих данных с применением современных методов машинного обучения. Для достижения цели необходимо решить следующие задачи:

1. Подготовить базу данных на основе разметки и существующих сведений о движениях на видеофайлах;
2. Подготовить обучающую (датасет) и контрольную выборки;
3. Обучить и протестировать созданную нейронную сеть.

В данной работе представлен обзор литературы и существующих исследований последних лет по теме, описываются задачи, связанные с распознаванием и предсказанием движений человека, а также ход действий и подробные шаги при подготовке датасета и обучении моделей.

Обзор существующих исследований. На момент написания работы существует множество исследований с задачей предсказания движений на видео с помощью методов глубокого обучения.

В [1] предлагается простой конвейер для классификации и локализации действий в видео без кадрирования. Авторы объединяют возможности как 3D-CNN (3D Convolutional Neural Network), так и RNN (3D Recurrent Neural Network) в единую структуру нейронной сети. Выходные данные из модели 3D-

CNN берутся в качестве входных данных для 3D-RNN. Сеть обрабатывает 16-кадровый клип с последовательностью признаков из видео и возвращает последовательность вероятностей классов. Модель обучается на датасете ActivityNet Challenge 2016 (640 часов видео). В ходе работы было достигнуто mAP 0,5874 и mAP 0,2237 в задачах классификации и локализации действий соответственно.

В работе [2] рассматривается проблема точной локализации ключевых кадров действия на видео. Авторы работы описали функции потерь, которые позволили бы уменьшить число ложноположительных прогнозов. Структурированная потеря основана на наилучшем совпадении между предсказанными и размеченными началами действий. Описаны вспомогательные функции: Matching Loss, Wasserstein/EMD Loss, Per-Frame Loss, Combined Loss. Используется рекуррентная нейронная сеть для минимизации структурированных потерь с помощью градиентного спуска. Работа функций проверяется на датасетах The Mouse Reach Dataset (видео начинается с того, что мышь тянется к таблетке, и заканчивается, когда она ее ест) и THUMOS’14 (большой набор видео с размеченными действиями). Для набора данных Mouse Reach было обнаружено, что Wasserstein/EMD Loss легче оптимизируется. Однако для набора данных THUMOS’14 Matching Loss показала наилучшие результаты.

В работе [3] авторы исследуют проблему обнаружения действий с помощью потоковых данных о скелете. Они предлагают многозадачную рекуррентную нейронную сеть Joint Classification-Regression для лучшего распознавания действия и его локализации. Используя совместно задачи классификации и регрессионной оптимизации, эта сеть способна автоматически более точно находить начало и конец действий. В частности, используя преимущества подсети deep Long Short-Term Memory (LSTM), предлагаемая модель автоматически фиксирует сложную долгосрочную временную динамику, которая избегает типичной конструкции скользящего окна и обеспечивает высокую вычислительную эффективность. Кроме того, подзадача регрессионной оптимизации дает возможность прогнозировать действие до его возникновения. Используется общедоступный набор данных, G3D (Gaming Action Dataset), на котором, по словам авторов, модель демонстрирует многообещающую производительность.

Авторы работы [4] решают задачу распознавания последовательностей человеческих действий из видеопотока. Цель – доказательство важности обнаружения начальной точки и впоследствии предложение метода определения начала текущего действия. Метод основан на двунаправленной нейронной сети (Bidirectional LSTM), которая вычисляет вероятность того, что кадр станет отправной точкой, сравнивая динамику действий до и после кадра. Эксперименты на трех наборах данных (основной датасет – Montalbano Gesture dataset) показали, что метод может надежно обнаружить начальную точку действия, улучшая точность раннего распознавания.

Задача распознавания и предсказания движений человека

Распознавание действий на видео по-прежнему остается сложной задачей из-за многих проблем, включая большую внутриклассовую разницу, нечеткую границу между классами, точку зрения, окклюзию (затенение), внешний вид, влияние окружающей среды и качества записи, в частности в реалистичных видео. Более того, чтобы иметь полноценную систему распознавания человеческих действий, необходимо учитывать другие дисциплины, например, психологию.

Распознавание позы человека – технология компьютерного зрения, которая основана на обнаружении и анализе положения человека в пространстве. Классическим подходом к эффективной оценке позы человека является использование каркаса, который визуально моделирует человеческое тело и его движения, представляя его как набор неких компонентов. Существует три наиболее часто используемых подхода: модель на основе скелета (skeleton-based), модель на основе контура (contour-based) и модель на основе объемных фигур (volume-based).

Скелетная модель основана на наборе координат ключевых точек, таких как лодыжки, колени, плечи, локти, запястья и прочие суставы, в комплексе составляющих скелетную структуру человеческого тела. Благодаря своей универсальности и гибкости эта модель используется как для 2D, так и для 3D методов распознавания позы. 2D-оценка основана на обнаружении и анализе координат (x,y) суставов человеческого тела на RGB-изображении. Для 3D-оценки добавляются координаты по оси z. В работе используется 2D-модель.

Контурная модель активно использовалась в прошлом и основывалась на контуре, полученном за счет деления частей туловища на прямоугольники и данных об их длине и ширине.

Наиболее простые объемно-ориентированные модели были аналогичны контурным, только вместо двумерных фигур для них использовались трехмерные (цилиндры, конусы и так далее). Современные модели обычно имеют вид сетки, полученной с помощью методов 3D-сканирования.

Технологические достижения в области компьютерных наук и инженерии позволяют компьютерным системам понимать человеческие действия, изображаемые на видео. В области компьютерного зрения существует две основные задачи, связанные с этим: распознавание и прогнозирование человеческих действий на основе полученных видеоданных. И если для распознавания действий необходимы данные, содержащие полный процесс действия, то прогнозирование действия основывается на выявлении причины его возникновения на основе неполных видео/сенсорных данных.

Действия часто оказываются типичными, выполняемыми в помещении, такими как ходьба, разговор, стояние и сидение. Они также могут быть более целенаправленными, такими как те, которые выполняются на кухне или на заводе. В набор видеоданных, принятых в данном препринте, были включены действия, которые могут быть частью работы на заводе автомобилестроения: подъем и опускание предметов (правильный и неправильный), работа в наклонном положении тела, над головой, над столом, и так далее.

Предсказание действия – это результат классификации неполных входных данных по действию, которое еще предстоит. Одной из подзадач является предвосхищение действия (action anticipation), то есть его распознавание, когда еще не наблюдается ни единого фрагмента действия, а классификация полностью основана на наблюдаемых контекстуальных подсказках. Другой – раннее предсказание действия по наблюдаемой его части (early action prediction). И то и другое является проблемами классификации, но предсказание часто требует набора данных с временными аннотациями и четкого деления между сегментом "до действия" и сегментом "во время действия" для предвосхищения действия или между "начальным действием" и "конечным действием" для раннего предсказания действия.

Временно́е предсказание действия (Temporal Activity Prediction) – это процесс разбиения входного видео на сегменты (последовательные серии кадров) действия и бездействия путем указания маркеров начала и конца каждого фрагмента действия. Локализация/обнаружение временно́го действия – это процесс создания предположения о временны́х рамках действия и классификации каждого действия. В нашей работе для всех видео существует разметка, где для каждого кадра отмечено, происходит в нем определенное действие или нет.

Существует множество фреймворков для работы с видеоданными. В препринте применяется подход с определением ключевых точек на основе распознавания 2D-координат на видео с помощью фреймворка OpenPose.

Нейронные сети для предсказания активности

В последнее десятилетие в исследованиях, связанных с распознаванием двигательной активности, произошел сдвиг парадигмы с преимущественно поверхностных, созданных вручную подходов к глубокому обучению. Теперь многослойные искусственные нейронные сети способны изучать сложные нелинейные отношения в структурированных данных.

Опишем несколько архитектур, которые могут быть использованы при работе над задачей прогнозирования типа физической активности.

Сверточные нейронные сети (CNN)

Ни один компонент архитектуры глубокого обучения не оказал бо́льшего влияния на понимание типа активности по изображениям (и на компьютерное зрение в целом), чем сверточные нейронные сети (CNN), также называемые

ConvNets. CNN в основном состоит из сверточных, объединенных, нормализованных и полностью связанных слоев. CNN полезны для понимания видео, поскольку совместное использование весов резко уменьшает число обучаемых параметров и, следовательно, снижает вычислительные затраты по сравнению с полностью подключенными сетями. Как правило, модели с большим числом слоев превосходят более мелкие модели за счет увеличения рецептивного поля - той части входного сигнала, которая вносит вклад в характеристику отдельных нейронов в сети. Однако глубокие модели могут страдать от таких проблем, как взрывающиеся или исчезающие градиенты.

Рекуррентные нейронные сети (RNN)

Второй по распространенности архитектурой искусственных нейронных сетей, используемой для понимания типа активности по изображениям, считается рекуррентная нейронная сеть (RNN). RNN используют подход с ориентированным графом для обработки последовательных входных данных, таких, как например, временны́е данные. Это делает их ценными для понимания активности, поскольку кадры (или извлеченные векторы на основе кадров) могут подаваться в качестве входных данных. Наиболее распространенным типом RNN являются сети долгой краткосрочной памяти LSTM (Long Short-Term Memory). Ячейка LSTM использует структуру входных/выходных ворот для выполнения долгосрочного обучения. Вторым наиболее распространенным типом RNN является закрытый рекуррентный блок GRU (Gated Recurrent Unit). Ячейка GRU использует структуру ворот сброса/обновления для выполнения менее интенсивного вычислительного обучения, чем ячейки LSTM.

Реализация программы

Код был написан на языке Python с использованием интерактивной среды Google Colab. Среда позволяет использовать основные современные возможности популярных библиотек **Python** для анализа и визуализации данных. В работе использованы такие библиотеки, как **numpy** для работы с массивами, **pandas** для обработки датасета, фреймворк **TensorFlow** и его надстройка, **Keras** для обучения нейросети и библиотека **mathplotlib** для визуализации данных с помощью графиков.

OpenPose – многопользовательская система [6], позволяющая обнаруживать ключевые точки человеческого тела, рук, лица и ног на отдельных изображениях. Фреймворк был выпущен в виде кода на языке программирования Python, реализации на C++ и плагина Unity.

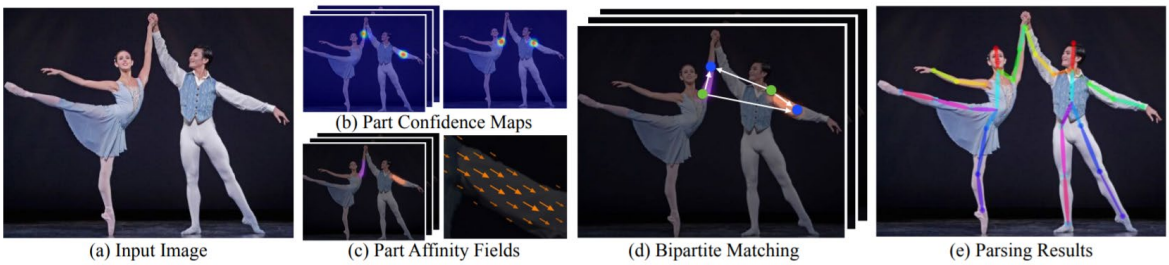

Рис. 1. Работа модуля OpenPose

Работа метода проиллюстрирована на (рис 1). Система принимает в качестве входных данных цветное изображение размером w×h и демонстрирует 2D-расположение анатомических ключевых точек для каждого человека на изображении.

Рис. 2. Работа системы **OpenPose** на собранных видеоданных

Описание датасета. В ходе лабораторного эксперимента по сбору данных было отснято порядка 10 часов материала с двух ракурсов. Все видеоданные были нарезаны на 5 минутные отрезки для удобства хранения.

Для каждого видео была создана разметка - таблица, в которой указывался номер кадра и шесть бинарных целевых переменных, соответствующих этому кадру, которые нужно предсказывать в задаче. Видео было записано с двух разных ракурсов. В каждой колонке стояла целевая переменная, которая соответствует этому действию: 1 – действие совершается, 0 – не совершается. Заполнение значений целевых переменных проводилось вручную посредством покадрового просмотра полученных данных с видео.

Таким образом, на входе было сформировано снятое с двух разных ракурсов видео, показывающее людей, которые выполняют шесть различных действий. На выходе имеются следующие бинарные метки:

1. Захват предмета
2. Подъем большого предмета
3. Опускание большого предмета
4. Работа над столом
5. Работа в наклоне
6. Работа над головой

Предобработка данных. Для обучения и тестирования нейронной сети необходимо было обработать видеофайлы, чтобы представить данные в нужном формате.

Датасет состоял из раскадрированного видео. На каждом полученном кадре **OpenPose** распознает характерные точки скелета и выводит информацию по 25 ключевым точкам в виде их координат по осям X, Y, а также данные о степени уверенности в правильном распознавании точек.

Для всех полученных видеоданных была проведена оценка правдоподобия распознанных программой OpenPose характерных точек скелетов. При этом использовалась не только фильтрация данных, но и эвристические алгоритмы, а также то, насколько естественными на видео кажутся действия и переходы между ними.

Фильтрация проводилась двумя методами. Первый исключал все плохо определенные точки, имеющие получаемую из OpenPose степень уверенности, равную нулю. Второй метод состоял в подборе изменяющихся пороговых значений для оставшихся используемых точек. Этот метод применялся в совокупности с интерполяцией по оставшимся точкам с оценкой степени уверенности их получения. Данные операции проводились по каждому измерению.

Эвристические алгоритмы включали в себя проверку моделей скелетов на правдоподобие. Применялось три метода. Первый оценивал пропорциональное сходство распознанных частей тела. Например, распознанное плечо и предплечье не должны отличаться по длине более чем на 20%, так как даже искажение перспективой не может слишком существенно повлиять на длину звеньев руки. Данный метод применялся как перекрестно между частями тела, так и для проверки симметрии частей тела. Второй метод проверял соответствие определенных точек на кадре: двух плечевых суставов; двух бедренных суставов; плечевого, тазобедренного суставов, точек на грудной клетке, тазовой кости. Третий метод оценивал гладкость движений человека посредством проверки скорости перемещения распознанных ключевых точек. Таким образом исключались выбросы или зашумленные точки выборки для последующей интерполяции.

Вручную в датасете были оставлены только максимально быстрые и естественные переходы между действиями.

Построение модели движения. Используется многоступенчатая структура сверточной нейронной сети (CNN). Такая сверточная модель делает прогнозы на основе истории фиксированной длительности, что может привести к лучшим результатам, чем более простые модели, поскольку она может оценивать изменения, происходящие со временем. При многоэтапном прогнозировании модель должна научиться предсказывать диапазон будущих значений. Таким образом, в отличие от одноступенчатой модели, где предсказывается только одна будущая точка, многоступенчатая модель предсказывает последовательность будущих значений.

Модель будет делать набор прогнозов на основе временного окна данных.

Основные особенности окон ввода:

1. Ширина - число временны́х шагов и меток;
2. Временно́й сдвиг между ними.

Создается программа WindowGenerator. Эта программа обрабатывает индексы и смещения, делит окна признаков на пары (features, labels), выстраивает содержимое полученных окон и генерирует пакеты этих окон из данных обучения, оценки и тестирования. Метод init включает в себя всю необходимую логику для обработки индексов ввода и меток. В качестве входных данных принимаются фреймы обучающей (70%), оценочной (20%) и тестовой (10%) выборок (рис. 4).

Учитывая список последовательных входных данных, метод split_window преобразует их в окно входных данных и окно меток (рис. 3).

```
def split_window(self, features):
  inputs = features[:, self.input_slice, :]
  labels = features[:, self.labels_slice, :]
  if self.label_columns is not None:
    labels = tf.stack(
        [labels[:, :, self.column_indices[name]] for name in self.label_columns]
        axis=-1)

  inputs.set_shape([None, self.input_width, None])
  labels.set_shape([None, self.label_width, None])

  return inputs, labels
```

Рис. 3. Функция преобразования входных данных

```
class WindowGenerator():
  def __init__(self, input_width, label_width, shift,
               train_df=train_df, val_df=val_df, test_df=test_df,
               label_columns=None):
    # Store the raw data.
    self.train_df = train_df
    self.val_df = val_df
    self.test_df = test_df

    # Work out the label column indices.
    self.label_columns = label_columns
    if label_columns is not None:
      self.label_columns_indices = {name: i for i, name in
                                    enumerate(label_columns)}
    self.column_indices = {name: i for i, name in
                           enumerate(train_df.columns)}
```

Рис. 4. Разделение датасета на выборки

Метод make_dataset преобразует датафрейм из (input_window, label_window) в tf.data.Dataset с помощью функции preprocessing.timeseries_dataset_from_array (рис.5).

```
def make_dataset(self, data):
  data = np.array(data, dtype=np.float32)
  ds = tf.keras.preprocessing.timeseries_dataset_from_array(
      data=data,
      targets=None,
      sequence_length=self.total_window_size,
      sequence_stride=1,
      shuffle=True,
      batch_size=32,)

  ds = ds.map(self.split_window)

  return ds
```

Рис. 5. Преобразование датасета в удобный формат

При многоэтапном прогнозировании модель должна научиться прогнозировать диапазон будущих значений. На рис. 6 представлена схема работы сети (для 30 значений на входе и выходе), на рис. 7 - реализация модели с шириной свертки, равной 5.

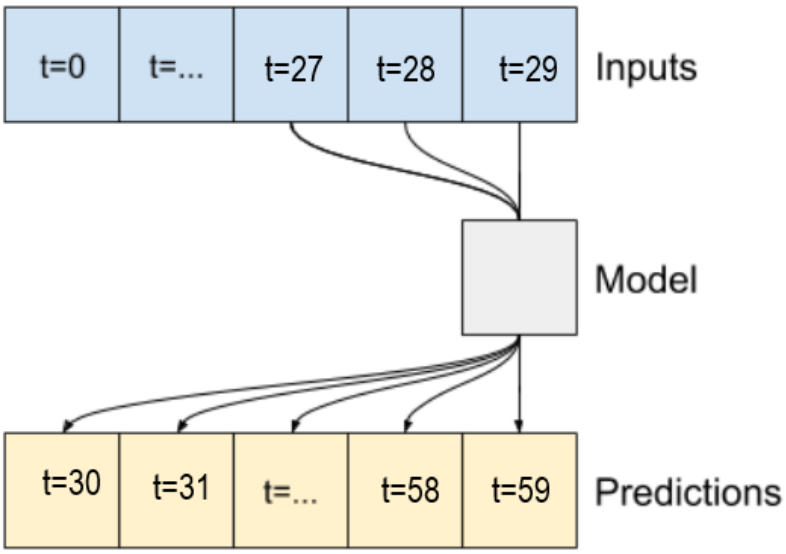

Рис. 6. Схема работы сети

```
CONV_WIDTH = 5
multi_conv_model = tf.keras.Sequential([
    # Shape [batch, time, features] => [batch, CONV_WIDTH, features]
    tf.keras.layers.Lambda(lambda x: x[:, -CONV_WIDTH:, :]),
    # Shape => [batch, 1, conv_units]
    tf.keras.layers.Conv1D(256, activation='relu', kernel_size=(CONV_WIDTH)),
    # Shape => [batch, 1,  out_steps*features]
    tf.keras.layers.Dense(OUT_STEPS*num_features,
                          kernel_initializer=tf.initializers.zeros()),
    # Shape => [batch, out_steps, features]
    tf.keras.layers.Reshape([OUT_STEPS, num_features])
])

history = compile_and_fit(multi_conv_model, multi_window)
```

Рис. 7. Реализация модели с шириной свертки, равной 5

Параметры модели

В качестве примера рассмотрим действие “захват предмета”, означающее, что человек на видео взаимодействует с каким-либо объектом. Был отснят 71 переход к действию, из них были сформированы 142 образца начала и конца действия. Так как действия были отсняты с двух ракурсов, то переходов в сумме оказалось 284. Каждый переход берется со своей временно́й окрестностью и называется слайсом. Длиной слайса называется радиус окрестности момента перехода действия. Все подходящие данные из всех видео объединяются в один датасет. Обучающий пример – это окно длинной X

кадров. Оно может содержать один, два перехода или может оказаться без переходов вовсе, все зависит от длины слайса и длины окна.

С целью достижения большей точности была выполнена серия процедур обучения с разными параметрами. При делении данных на слайсы размером в 15 кадров (половина секунды до перехода и после него) получаем датасет с 76 столбцами (75 – данные OpenPose, 1 – столбец разметки для предсказываемого действия “захват предмета”). При длине окна в 10 кадров результаты неудовлетворительны: ошибка определения кадра начала действия по метрике *accuracy* составляет 0.571. На рис.7 зелеными точками обозначены поданные данные, красными крестиками – предсказания модели. Перепад графика означает переход от нуля к единице или наоборот, то есть окончание или начало действия. Модель ожидает переход там, где его нет.

Практически аналогичны прогнозы для размера окна, равного 12 (точность равна 0.5781) и 15 (точность равна 0.6115). Таким образом, ясно, что размер слайса-примера должен быть больше.

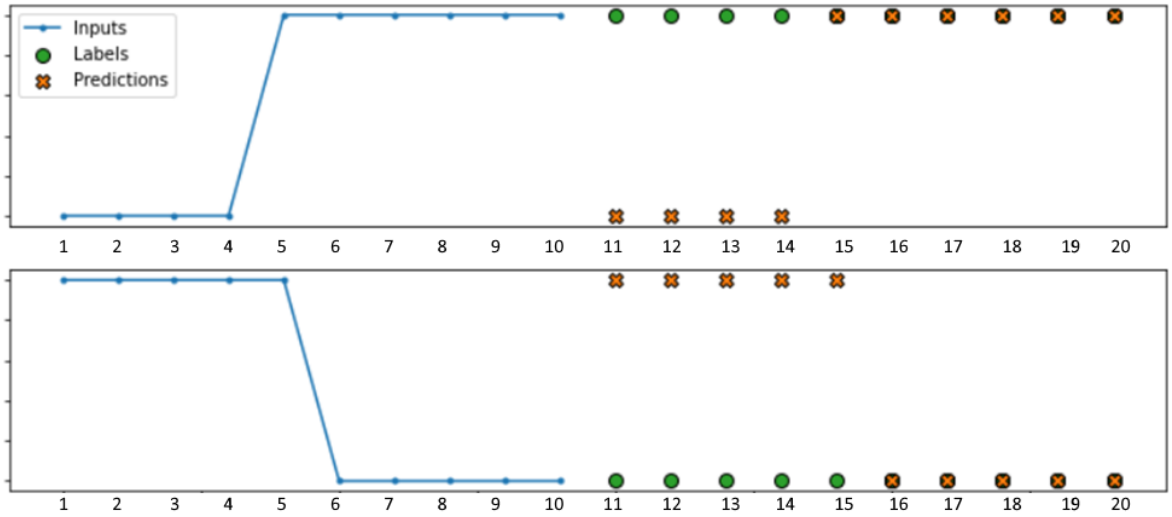

Рис. 7. Пример работы модели по определению действий по 10 кадрам

При увеличении окна до 20 кадров точность падает до 0.5984, при 30 кадрах точность равна 0.6060, переходы в обоих случаях распознаются в местах, где их нет (рис. 8).

При увеличении размера слайсов до 60 кадров (две секунды), результаты обучения улучшаются существенно. Отметим, что благодаря расширению окон увеличивается число положительных примеров, так как считаются все возможные примеры, где точка перехода попадает во внутренние кадры.

При размере окна в 20 кадров точность обучения возрастает до 0.7262. При тестировании работы сети окном по 30 кадров на слайсах длиной в 60 кадров результат предсказания вырастает до 0.8676 (рис. 9). В первом примере модель ошиблась на 16 кадров, но абсолютно верно обнаружила переход во

втором примере и всего на один кадр ошиблась в третьем, четвертом, пятом и на два кадра в шестом.

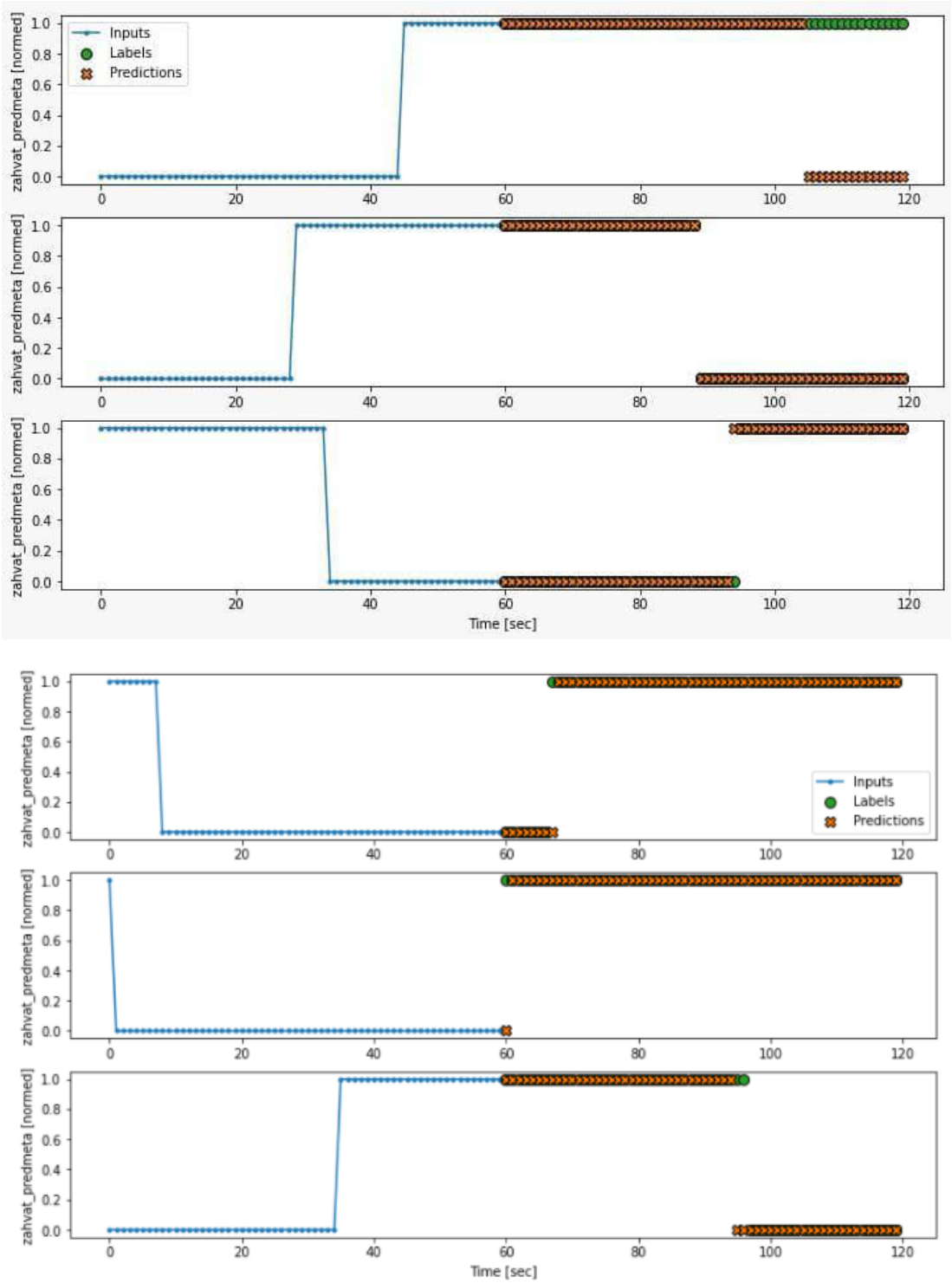

Рис. 8. Пример работы модели по определению действий по 60 кадрам

Результаты

В результате самой эффективной оказывается нарезка видеоданных на слайсы размером в 60 кадров и подача данных в модель сети окном по 30 кадров (Таблица 1).

Заключение

В препринте представлен обзор литературы и существующих исследований последних лет по теме исследования, описаны актуальные задачи, связанные с распознаванием и предсказанием движений человека, а также ход действий и подробные шаги при подготовке датасета и обучении модели.

Сконструирован и реализован метод предсказания физической активности человека с помощью нейронной сети. В качестве входных данных в нейронную сеть была использована разметка видеоданных, на которых были зафиксированы возможные действия рабочего на предприятии. В результате была создана сверточная нейронная сеть, способная при определенных параметрах, которые были подобраны экспериментально, предсказывать действия и обнаруживать переходы между ними с точностью до 87%. Итоговая оценка качества работы сети представлена в таблице 1. В дальнейшем возможно использование результатов исследования для более продвинутого взаимодействия с ассистирующими устройствами на предприятиях.

Таблица 1

Результаты анализа модели.

Размер слайсов	Размер окна	Точность
15	10	0.5715
15	12	0.5781
15	15	0.6115
30	10	0.6279
30	20	0.5984
30	30	0.6060
60	20	0.7262
60	30	0.8676
60	60	0.6500

Библиографический список

1. Montes A. Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks, 2017.
2. Iljung S. Kwak. Detecting the Starting Frame of Actions in Video, 2020.
3. Li Y. Online Human Action Detection using Joint Classification-Regression Recurrent Neural Networks, 2016.
4. Wang B. Back to the beginning: Starting point detection for early recognition of ongoing human actions, 2018.
5. Hutchinson M., Gadepally V. Video Action Understanding: A Tutorial, 2020.
6. Cao Z [и др]. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, 2019.
7. Pham H.H. [и др]. Video-based human action recognition using deep learning: a review, 2019.
8. Chena Y., Tianb Y., Hea M. Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods, 2020.
9. Podoprosvetov, A.V., Alisejchik, A.P., Orlov, I.A. Comparison of action recognition from video and IMUs Procedia Computer Science, 2021, 186, стр. 242–249

Оглавление