МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет Механико-математический

Кафедра Теоретической кибернетики

Направление подготовки 02.03.01. - Математика и компьютерные науки

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Мукумов Анвар Рустамович

(Фамилия, Имя, Отчество автора)

Тема работы Распознавание слов жестового языка по видеофрагментам

«К защите допущен»

Заведующий кафедрой,

д.ф.-м.н., профессор

Ерзин А. И. .....

(фамилия , И., О.) (подпись, МП)

«03»..... июня .....2024 г.

Научный руководитель

к.ф-м.н., доцент,

доцент,

Неделько В. М. .....

(фамилия , И., О.) (подпись, МП)

«31»..... мая .....2024 г.

Дата защиты: «20»..... июня .....2024г.

Новосибирск, 2024

Оглавление

Введение

Жестовый язык - это визуальный способ коммуникации, используемый людьми с нарушениями слуха или речи, а также их семьями, друзьями и коллегами. В России, около 13 млн. человек имеют нарушения слуха, а 300 тыс. - полностью неслышащие, и для них общение с людьми, не знающими русского жестового языка вызывает сложности и неудобства. В связи с этим, исследование в области автоматического распознавания языка жестов поможет улучшить коммуникацию слабослышащих людей с окружающими.

Также, в наше время, нейросетевые алгоритмы показывают себя как мощный инструмент для решения различных задач по обработке изображений, текстов и последовательностей, превосходящий по качеству альтернативные алгоритмы машинного обучения, в связи с чем являются более предпочтительными для решения задач со сложной структурой данных.

Таким образом, разработка нейросетевого решения для классификации фиксированного набора слов русского жестового языка по видеофрагментам является актуальной задачей.

Целью данной работы является разработка алгоритма решения задачи классификации слов русского жестового языка при помощи нейронных сетей, с входными данными - координатами ключевых точек кистей рук, полученных из видеофрагментов с помощью нейронной сети MediaPipe и анализ полученных результатов.

В области распознавания слов языка жестов уже имеются некоторые исследования. Решение задачи с исследуемым набором данных рассмотрено в статье [1], в которой создатели рассматриваемого набора данных, помимо описания сбора видеофрагментов, предоставили результаты решения задачи классификации жестов по ним. В данной работе были использованы три архитектуры нейронных сетей - ResNet3D-50, Swin-Large и MViTv2. Все три архитектуры были предобучены на наборе данных Kinetics [2], в котором представлены 500000 видеофрагментов с 600 различными действиями, совершаемыми людьми. Далее, архитектуры дообучались на разбитых на кадры видеофрагментах. В результате, данные нейросети показали значения точности классификации: 43.9 %, 55,6% и 64% соответственно. Однако, в исследовании не была рассмотрена возможность автоматической разметки ключевых точек на руках при помощи нейронной сети, а затем обучения и формирования прогнозов на основе этих данных.

Также, есть исследование [3] в области распознавания американского языка жестов, однако в нем рассмотрен гораздо больший объем данных на класс, и для обучения используются координаты ключевых точек не только кистей, но и губ, носа, и некоторых других частей тела. Всего используется 1662 ключевые точки. Также, рассмотрено всего 10 классов, при выборке объема 750.

В ходе работы было построено решение задачи с использованием преимуществ рекуррентных нейронных сетей и кодирующих слоев

трансформеров. Полученное решение имеет качество классификации 44,3% на тестовой выборке при имеющихся 1000 классах, 15 обучающих, и 5 тестовых примерах на каждый класс. В проведенном исследовании, распознавание жестов происходит только по координатам ключевых точек кистей рук, в отличие от остальных имеющихся исследований этой и похожих задач. Преимуществом данного подхода является то, что полученное решение не зависит от наличия в кадре каких-либо частей тела, кроме кистей рук, в отличие от других решений, в которых алгоритм опирается на другие части тела, в частности, обращает внимание на подсказки, такие как эмоции, испытываемые диктором и произнесение слов губами.

Таким образом, полученное в данном исследовании решение является более универсальным, а также требует гораздо меньших вычислительных мощностей, объемов памяти и времени, и при этом превосходит по качеству одно из трех существующих решений, опирающихся на подход, рассматривающий гораздо большее количество дополнительной информации, и способно конкурировать с решениями, имеющими более высокую точность ввиду их неустойчивости к отсутствию дополнительной информации в кадре.

1. Построение алгоритма классификации слов жестового языка по координатам ключевых точек на кистях рук.

В данной главе приводится математическая постановка задачи, обзор исследуемого набора данных, архитектуры используемых для решения нейронных сетей, оптимизируемые функция потерь и целевая метрика и алгоритм оптимизации.

1.1. Математическая постановка задачи

Имеется X- пространство трехмерных объектов $\{x_{ijk}\}$, $i = \overline{1,\ 3}$, $j = \overline{1,\ 42}$, $k = \overline{1,\ n(x)}$, координат 42-х ключевых точек на видеофрагменте длины $n(x)$ кадров; Y - множество меток классов (слов русского жестового языка). Из X извлечена некоторая выборка S с известными метками классов.

На основе имеющихся данных, требуется построить функцию $f : X \to Y$, сопоставляющую элементам из X метки классов из Y наилучшим, в некотором смысле, образом. В качестве критерия рассматривается значение выбранной метрики точности классификации на отложенной подвыборке из S

1.2. Обзор набора данных

В данной работе, задача решается на наборе данных “Slovo - Russian Sign Language Dataset” [1, 4]. В наборе имеется 20000 видеофрагментов, где люди показывают жестами по одному слову русского языка на видео, таблица с идентификаторами видеофайлов и пользователей, и переводом жестов в виде текста. Также имеется файл, содержащий информацию о координатах ключевых точек на данных видеофрагментах, полученных при помощи нейросети MediaPipe. Всего имеется 1000 уникальных слов, и на каждое из них по 20 видеофрагментов. Данные поделены на обучающую и валидационную выборку в соотношении 3:1, с сохранением баланса классов.

Рассматриваемое координатное пространство является трехмерным и имеет следующие компоненты: x - горизонтальная, y - вертикальная, z - мера глубины. Координаты x и y нормализованы, так как разрешения видеофрагментов различаются. Для видео размера $(W \times H)$: $x = \frac{x'}{W}, y = \frac{y'}{H}$. Для обучения модели, слова были закодированы числами от 0 до 1000.

Также, в одном из экспериментов, модель была предобучена на наборе данных со словами американского языка жестов [5]. Из набора данных были извлечены только координаты тех же ключевых точек на кистях рук, которые были использованы в наборе данных Slovo [1, 4].

Всего в наборе данных имеется 95000 последовательностей координат ключевых точек, относимых к 250 классам. Для предобучения, данные были поделены на обучающую и тестовую выборку в отношении 3:1 с сохранением баланса классов.

1.3. Архитектура модели

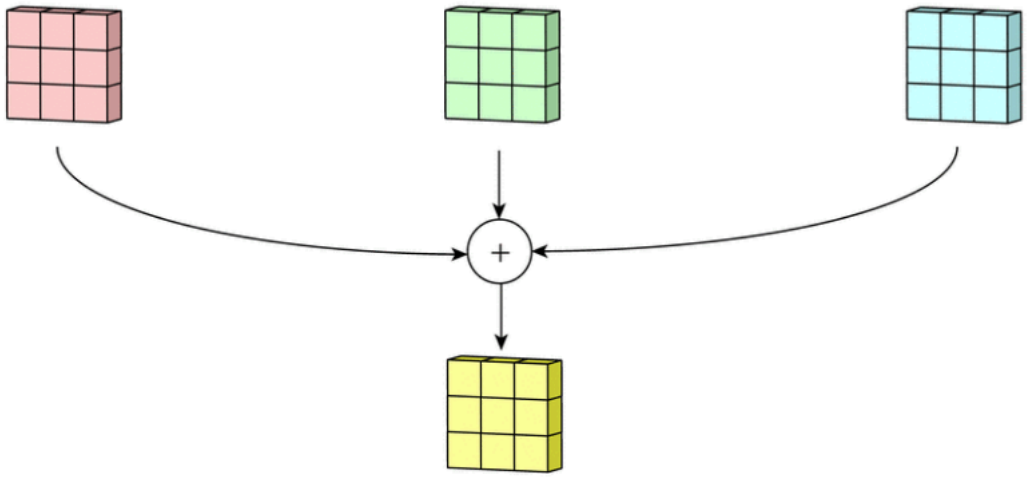
Данные представляют собой последовательности, в связи с чем было принято решение рассмотреть рекуррентные нейронные сети для того, чтобы учесть зависимость координат между кадрами. Отличительная особенность рекуррентных нейронных сетей в том, что во время обучения нейроны отправляют сигнал не только дальше по сети, но и в обратном направлении. Таким образом, рекуррентная нейронная сеть учитывает взаимосвязь признаков в разные моменты времени.

В качестве представителя рекуррентной нейронной сети было решено использовать LSTM. Ее отличие от других рекуррентных нейронных сетей состоит в том, что по мере обучения более ранние наблюдения “забываются”, имеют меньший вес.

В качестве механизма генерации высокоуровневых признаков были применены одномерные сверточные слои с группировкой по каналам, с пакетной нормализацией и функцией активации ReLU.

$$ReLU(x) = max(0, x)$$

Для одного канала работа сверточного слоя основана на том, что он инициализирует некоторую матрицу W - ядро свертки, заранее заданного размера. Затем осуществляется проход по входной матрице частями того же размера, что и W, и к этим фрагментам применяется операция линейной свертки с матрицей W, и таким образом получаются элементы выходной матрицы. В случае C > 1 входных каналов, ядро свертки представляет собой C матриц, которые проделывают ту же процедуру, каждая по своему каналу, а затем результаты суммируются поэлементно (рисунок 1). Также, в качестве параметра сверточного слоя задается число выходных каналов C'. В случае, когда $C' > 1$, описанная выше процедура повторяется C' раз, затем результаты объединяются в выходной тензор, посредством конкатенации C' полученных матриц по размерности каналов. Однако, в случае группировки по каналам, сначала входной тензор делится на k равных частей вдоль размерности каналов, затем сверточный слой применяется к каждой части по отдельности с количеством выходных каналов C', затем результаты объединяются в выходной тензор с количеством каналов kC'. Таким образом, при группировке по каналам, взаимодействовать будут только кадры, находящиеся близко друг к другу, в связи с чем данные все еще будут представлять из себя последовательности, но уже агрегированные по небольшим окнам, поэтому LSTM все еще остается актуальной, в отличие от ситуации, в которой сверточные слои влияли бы сразу на всю последовательность.



(рисунок 1)

Также, в последнее время большую популярность имеет применение трансформеров к задачам, связанным с обработкой последовательностей, и, как показал эксперимент, применение кодирующих слоев совместно со сверточными слоями улучшает результат. Главная особенность трансформеров - механизм внутреннего внимания. Он устроен так, что при кодировании элемента последовательности учитывается то, насколько важен каждый из остальных элементов по отношению к кодируемому при помощи расчета скалярных произведений векторов отнесенных к разным элементам последовательности, полученных умножением векторов из входной последовательности на специальные матрицы запроса, ключа и значения.

Данный подход впервые был применен к текстовым данным, так как имеет хорошую интерпретацию с зависимостью слов в предложении, однако в настоящее время хорошо показывает себя при работе с

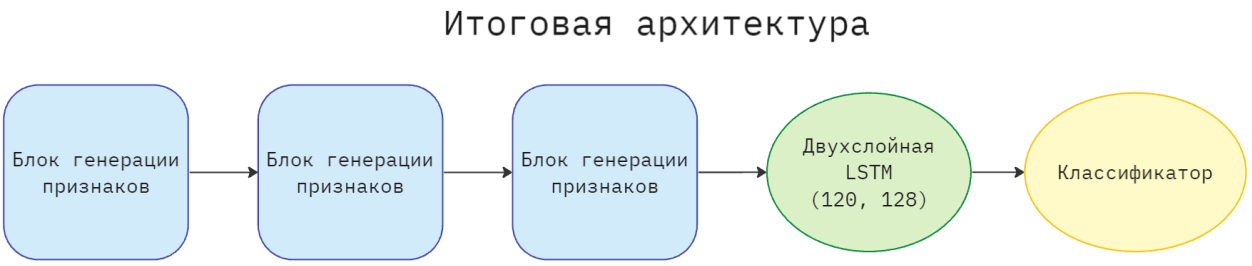
последовательностями другой природы, а также с изображениями. Более подробно данный механизм описан в статье [6].

Итоговая архитектура имеет следующий вид: три блока генерации признаков, состоящие из последовательного применения нескольких сверточных слоев с группировкой по каналам и кодирующих слоев трансформера, двухслойная LSTM и два полносвязных слоя с функцией активации ReLU между ними. Внутри блоков генерации признаков также применяется пакетная нормализация и функция активации ReLU.

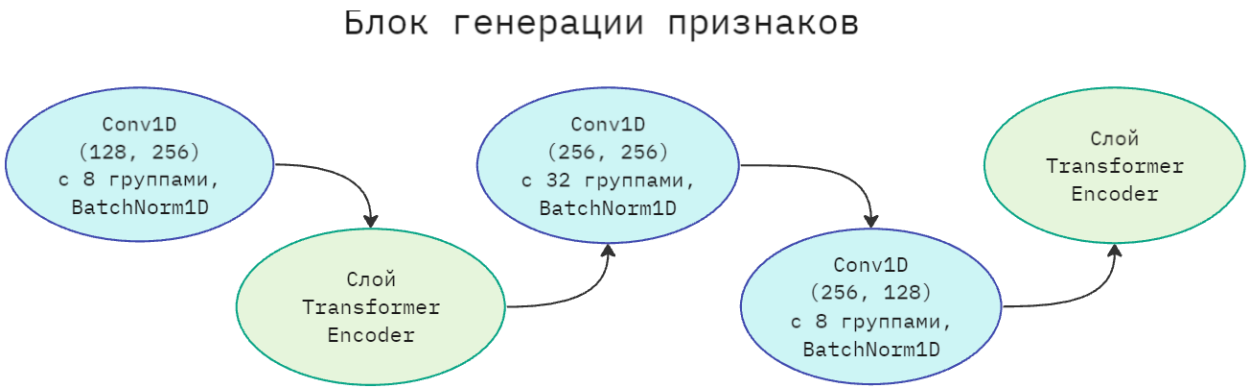
Для препятствования переобучению, используется регуляризация при помощи механизма случайного отключения нейронов в LSTM и в кодирующих слоях. Действует он следующим образом: при обучении каждый нейрон в слое с некоторой заданной вероятностью имеет выход, равный нулю, а в режиме тестирования модели, выходу с этого нейрона присваивается вес, равный частоте его отключения при обучении. Таким образом, нейронная сеть будет обучаться более равномерно и не будет концентрироваться на весах отдельных групп нейронов.

К выходному вектору значений применяется функция Softmax, дающая оценку распределения вероятности классов. Softmax от вектора z вычисляется по следующей формуле:

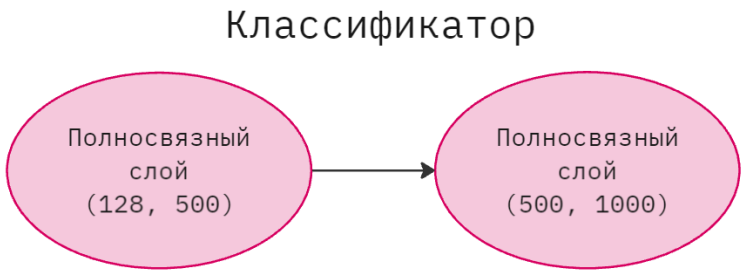
$$Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$



(рисунок 2)



(рисунок 3)



(рисунок 4)

На данных рисунках, conv1D означает сверточный слой с одной пространственной размерностью, а BatchNorm1D - пакетную нормализацию. То есть для каждого пакета данных считается текущее среднее значение и стандартное отклонение в каждом нейроне, а затем данные нормализуются в каждом нейроне без изменения размеров входа.

Также, для сверточного слоя указано число входных и выходных каналов и число групп для группировки по каналам. Для полносвязных слоев и для LSTM указаны размеры входного и выходного вектора. Двухслойный LSTM означает 2 одинаковых слоя LSTM, в которых выход с одного слоя поступает во второй. При переходе от одного слоя к другому используется функция активации ReLU.

1.4. Функция потерь

В качестве функции потерь была выбрана перекрестная энтропия. Ее значение на одном объекте вычисляется по следующей формуле:

$$H(y, p) = -\sum_{i=1}^{K} I\{y = t_i\} * log(p_i)$$

где K - количество классов, t - множество меток классов, p_i - вероятность объекта принадлежать классу i. Оценкой вероятности p_i является i-я компонента функции SoftMax от выхода с нейронной сети. Слой SoftMax преобразует выход так, чтобы он давал в сумме единицу. Далее, значение перекрестной энтропии суммируется по объектам и делится на их количество. Мотивацией применения данной функции потерь является то, что классы имеют идеальный баланс, а значит функция потерь с различными весами для каждого класса не требуется.

1.5. Целевая метрика и алгоритм оптимизации

В качестве целевой метрики, в силу сбалансированности классов была выбрана точность:

$$Accuracy = \frac{Correct}{N},$$

где Correct - число верных предсказаний, а N - общее число предсказаний.

Нейросеть оптимизируется с помощью алгоритма обратного распространения ошибки, с помощью оптимизатора Adam с градиентным шагом 0.0001 и планировщиком градиентного шага, умножающим его на 0.8 каждые 10 эпох, где эпоха - один полный проход по обучающей выборке.

Также были проведены эксперименты со стохастическим градиентным спуском в качестве оптимизатора и с другими градиентными шагами и параметрами планировщика, однако точность классификации уступала полученной при описанных выше параметрах.

2. Предобработка и преобразование данных.

В данной главе будут представлены использованные методы предобработки входных данных и их аугментации.

2.1. Предобработка данных

Координаты были получены в нормализованном виде, поэтому дополнительной нормализации не требовалось. Ко всем данным были применены следующие преобразования:

- Все отсутствующие значения координат были заменены нулями;
- Все последовательности длиннее 128 были обрезаны до данного размера;
- Все последовательности короче 128 были дополнены нулями до данного размера;
- Полученные данные были преобразованы из вида [3, 42, 128] к плоскому виду: [126, 128] с целью использования LSTM.

2.2. Аугментации

Обучающая выборка имеет очень маленький размер, в связи с чем, очень важно применить аугментации для того чтобы ее расширить и увеличить

обобщающую способность нейросетевой модели и предотвратить переобучение

В ходе решения задачи были применены следующие аугментации:

- Поворот вокруг начала координат на случайный угол из отрезка [-0.3, 0.3] радиан с вероятностью p = 0.35;
- Сдвиг на случайный вектор, с вероятностью p = 0.35, координаты вектора сдвига варьируются от -0.5 до 0.5 в нормализованном виде;
- Отражение относительно вертикальной оси с вероятностью p = 0.5

Важным замечанием является то, что к имеющимся координатам нельзя применять поворот без преобразования их к изначальному виду. Для корректности преобразования, координаты были умножены на соответствующие им компоненты разрешения видео, из которого они были извлечены и только после этого умножены на матрицу поворота, а затем возвращены к нормализованному виду, для того чтобы исключить влияние различных разрешений видеофрагментов.

Все эти аугментации несут некоторый смысл. Повороты и сдвиги помогают модели адаптироваться к тому, что люди могут находиться в разных областях кадра на разных видео, а отражение позволит модели продолжать хорошо работать на видео, снятых и на заднюю камеру, и на переднюю.

Аугментации применяются на каждой итерации извлечения объекта выборки с указанными выше вероятностями.

3. Предобучение.

С целью повышения качества классификации, построенная архитектура была предобучена на наборе данных со словами американского языка жестов [5]. К данным была применена та же предобработка, что была описана в главе 2. За 200 эпох обучения была получена точность 73.8%. В качестве входных данных не использовались координаты ключевых точек губ, носа и других частей лица и тела, помимо кистей рук и несмотря на это получена хорошая точность классификации. Затем, к предобученной нейронной сети был добавлен полносвязный слой для увеличения размера выходного вектора до 1000 и вся нейронная сеть была дообучена на рассматриваемом в данной работе наборе данных.

В результате данного эксперимента, точность классификации заметно увеличилась, с 39.6% до 44.3%. Данный результат показывает полезность предобучения на сторонних похожих наборах данных, а также то что построенная архитектура имеет потенциал к повышению качества классификации при увеличении объема обучающей выборки.

4. Результаты экспериментов.

Архитектура нейронной сети подбиралась при помощи наблюдения за изменением целевой метрики при ее изменении. Модели, в которых использовалась и рекуррентная нейронная сеть, и механизм внимания показали себя лучше всего. Также, точность классификации значительно растет при применении аугментаций и предобучения на данных, похожих на рассматриваемые по своей природе.

Обучение производилось на облачных сервисах kaggle.com на GPU NVIDIA TESLA P100, google colab на GPU NVIDIA TESLA T4 и yandex datasphere на GPU NVIDIA TESLA V100.

В результате проведенных экспериментов в среднем, были получены следующие значения точности классификации:

Число эпох	**Модель без аугментаций**	**Модель с аугментациями**	**Предобученная модель**
100	0.193	0.322	0.413
200	0.288	0.368	0.421
300	0.292	0.396	0.443

(таблица 1)

5. Анализ результатов.

Эксперимент показал, что применение аугментаций существенно улучшает выбранную метрику качества классификации.

Предобучение разработанной модели на наборе данных для распознавания американского языка жестов, позволило заметно повысить качество классификации. Получено значение точности 44.3% при имеющихся решениях с другим подходом с точностью 43.9 %, 55,6% и 64%. Однако имеющиеся в общем доступе решения имеют ряд недостатков:

Во-первых, обучение на самих видеофрагментах вычислительно тяжелая задача. Есть два возможных подхода к обучению такой модели:

1) Сохранение видеофрагментов в виде последовательности кадров в оперативной памяти,
2) Извлечение кадров в процессе обучения.

В первом случае, требуются очень большие ресурсы оперативной памяти, а во втором случае, требуются большие затраты времени. В обоих случаях, решение данной задачи становится невозможным при использовании общедоступных платформ для машинного обучения и анализа данных, таких как kaggle.com, google colab и yandex datasphere, на которых проводилось обучение архитектуры нейронной сети, разработанной в данном исследовании.

Во-вторых, решение, основанное на координатах ключевых точек кистей рук более универсально, так как модель будет обращать внимание

исключительно на движение рук и будет устойчива к отсутствию эмоций диктора и проговаривания им показываемых слов губами, а также в принципе отсутствию лица в кадре.

Заключение

В ходе данной работы было построено решение задачи классификации слов языка жестов, принимающее в качестве входных данных исключительно координаты ключевых точек на руках, предварительно извлеченных из видеофрагментов при помощи нейронной сети MediaPipe.

Экспериментально была подобрана архитектура, использующая механизм внутреннего внимания, особенности рекуррентных слоев и сверточных слоев.

Применены аугментации, повысившие качество классификации построенного алгоритма.

Нейросеть была предобучена на стороннем наборе данных, что помогло повысить качество классификации

Было проведено сравнение итоговой точности в 44.3% с имеющимися результатами в 43.9 %, 55,6% и 64%.

Была обоснована полезность данного подхода при уступающей двум из трех имеющихся решений точности, ввиду универсальности решения, гораздо меньших требований к вычислительным мощностям, а также ввиду того, что точность 44.3% является хорошим результатом для задачи классификации с большим количеством классов и малым количеством обучающих примеров.

Источники информации

[1] Kapitanov Alexander, Kvanchiani Karina, Nagaev Alexander, Petrova Elizaveta Slovo: Russian Sign Language Dataset. //International Conference on Computer Vision Systems (ICVS), 2023, pp. 63-76,

[2] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, Andrew Zisserman The Kinetics Human Action Video Dataset // preprint arXiv:1705.06950 [cs. CV] 19 May 2017

[3] Ahmed Mateen Buttar, Usama Ahmad, Abdu H. Gumaei, Adel Assiri, Muhammad Azeem Akbar, Bader Fahad Alkhamees Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs // MDPI open access journal Mathematics Volume 11 Issue 17

[4] Kapitanov Alexander, Kvanchiani Karina, Nagaev Alexander, Petrova Elizaveta Slovo - Russian Sign Language Dataset // Kaggle URL: https://www.kaggle.com/datasets/kapitanov/slovo

License: CC BY-SA 4.0

[5] Ashley Chow, Glenn Cameron, Mark Sherwood, Phil Culliton, Sam Sepah, Sohier Dane, Thad Starner Google - Isolated Sign Language Recognition // Kaggle URL: https://kaggle.com/competitions/asl-signs

License: CC BY 4.0

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin Attention Is All You Need // 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. arXiv:1706.03762v7 [cs.CL] 2 Aug 2023