

Отчет. Лабораторная работа №3.

ПОСТАНОВКА ЗАДАЧИ

В лабораторной работе №3 в качестве входных данных берется GTSRB - немецкий тестовый набор изображений дорожных знаков для распознавания. Ставится задача - разработать полносвязную нейронную сеть с использованием одной из библиотек глубокого обучения для решения задачи **Распознавания дорожных знаков**. Цель экспериментов - поиск оптимальных параметров настройки сети.

Таковыми параметрами являются:

- Количество слоев
- Количество эпох
- Функция активаций в скрытых слоях
- Функция инициализации весов в скрытых слоях
- Функция оптимизации
- Размер ядра свертки
- Количество входных фильтров в свертке

Классификация

Задача классификации — это задача присвоения меток объектам. Например, если объекты — это фотографии, то метками может быть содержание фотографий: содержит или изображение пешехода или нет, изображен ли мужчина или женщина, какой породы собака изображена на фотографии. Обычно есть набор взаимоисключающих меток и сборник объектов, для которых эти метки известны. Имея такую коллекцию данных, необходимо автоматически расставлять метки на произвольных объектах того же типа, что были в изначальной коллекции. Давайте формализуем это определение.

Допустим, есть множество объектов X . Это могут быть точки на плоскости, рукописные цифры, фотографии или музыкальные произведения. Допустим также, что есть конечное множество меток Y . Эти метки могут быть пронумерованы. Мы будем отождествлять метки и их номера. Таким образом $Y = \{red, green, blue\}$ в нашей нотации будет обозначаться как $Y = \{1, 2, 3\}$. Если $Y = \{0, 1\}$, то задача называется задачей бинарной классификации, если меток больше двух, то обычно говорят, что это просто задача классификации. Дополнительно, у нас есть входная выборка $D = \{(x_i, y_i), x_i \in X, y_i \in Y, i = [1, ..., N]\}$. Это те самые размеченные примеры, на которых мы и будем обучаться проставлять метки автоматически. Так как мы не знаем классов всех объектов точно, мы считаем, что класс объекта — это случайная величина, которую мы для простоты тоже будем обозначать. Например, фотография собаки может классифицироваться как собака с вероятностью 0.99 и как кошка с

вероятностью 0.01. Таким образом, чтобы классифицировать объект, нам нужно знать условное распределение этой случайной величины на этом объекте $p(y|x)$.

Задача нахождения $p(y|x)$ при данном множестве меток Y и данном наборе размеченных примеров $D = \{(x_i, y_i), x_i \in X, y_i \in Y, i = [1, \dots, N]\}$ называется задачей классификации.

Вероятностная постановка задачи классификации

Чтобы решить эту задачу, удобно переформулировать ее на вероятностном языке. Итак, есть множество объектов X и множество меток Y . $\xi : \Omega \rightarrow X$ — случайная величина, представляющая собой случайный объект из X . $\eta : \Omega \rightarrow Y$ — случайная величина, представляющая собой случайную метку из Y . Рассмотрим случайную величину $(\xi, \eta) : \Omega \rightarrow (X, Y)$ с распределением $p(x, y)$, которое является совместным распределением объектов и их классов. Тогда, размеченная выборка — это сэмплы из этого распределения $(x_i, y_i) \sim p(x, y)$. Мы будем предполагать, что все сэмплы независимо и одинаково распределены.

Задача классификации теперь может быть переформулирована как задача нахождения $p(y|x)$ при данном сэмпле $D = \{(x_i, y_i), x_i \in X, y_i \in Y, i = [1, \dots, N]\}$.

ТРЕНИРОВОЧНЫЕ И ТЕСТОВЫЕ НАБОРЫ ДАННЫХ



Характеристики примеров






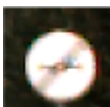

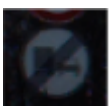

Количество классов	43
Максимальная ширина	243
Максимальная высота	225
Минимальная ширина, высота	25
Медиана для ширины, высоты	43

Распределение количества изображений по классам

Для тренировочного набора данных:

Номер класса и описание	Количество изображений	Пример изображения	Номер класса	Количество изображений	Пример изображения
0. Ограничение скорости 20	210		22. Неровная дорога	390	
1. Ограничение скорости 30	2220		23. Опасная обочина	510	
2. Ограничение скорости 50	2250		24. Сужение дороги	270	
3. Ограничение скорости 60	1410		25. Дорожные работы	1500	
4. Ограничение скорости 70	1980		26. Светофорное регулирование	600	

5. Ограничение скорости 80	1860		27. Пешеходный переход	240	
6. Конец участка с ограничением скорости	420		28. Дети	540	
7. Ограничение скорости 100	1440		29. Пересечение с велосипедной дорожкой	270	
8. Ограничение скорости 120	1410		30. Снег или лед	450	
9. Обгон запрещен	1470		31. Дикие животные	780	
10. Обгон грузовым автомобилям запрещен	2010		32. Конец зоны всех ограничений	240	
11. Пересечение со второстепенной дорогой	1320		33. Движение направо	689	
12. Главная дорога	2100		34. Движение налево	420	
13. Уступите дорогу	2160		35. Движение прямо	1200	
14. Движение без остановки запрещено	780		36. Движение прямо или направо	390	
15. Движение запрещено	630		37. Движение прямо или налево	210	
16. Движение грузовых автомобилей запрещено	420		38. Объезд препятствия справа	2070	

17. Въезд запрещен	1110		39. Объезд препятствия слева	300	
18. Прочие опасности	1200		40. Круговое движение	360	
19. Опасный поворот налево	210		41. Конец зоны запрещения обгона	240	
20. Опасный поворот направо	360		42. Конец зоны запрещения обгона грузовым автомобилям	240	
21. Опасные повороты	330				

Для тестового набора данных:

Номер класса	Количество изображений	Номер класса	Количество изображений	Номер класса	Количество изображений
0	60	15	210	30	150
1	720	16	150	31	270
2	750	17	360	32	60
3	450	18	390	33	210
4	660	19	60	34	120
5	630	20	90	35	390
6	150	21	90	36	120
7	450	22	120	37	60
8	450	23	150	38	690
9	480	24	90	39	90
10	660	25	480	40	90
11	420	26	180	41	90
12	690	27	60	42	90
13	720	28	150		
14	270	29	90		

МЕТРИКА КАЧЕСТВА РЕШЕНИЯ ЗАДАЧИ

За **метрику качества решения** взята точность (ассигасу). Отношение правильно определенных дорожных знаков к общему числу изображений подаваемых на вход нейронной сети.

ИСХОДНЫЙ ФОРМАТ ХРАНЕНИЯ ДАННЫХ

Данные представляют собой набор 3-канальных изображений имеющие набор параметров

№	Path	Width	Height	Roi.X1	Roi.Y1	Roi.X2	Roi.Y2	ClassId
0	Test/00000.png	53	54	6	5	48	49	16
1	Test/00001.png	42	45	5	5	36	40	1
2	Test/00002.png	48	52	6	6	43	47	38
3	Test/00003.png	27	29	5	5	22	24	33
n

Roi - отступы от левого верхнего угла для определения положения дорожного знака на изображении.

ФОРМАТ, В КОТОРОМ ДАННЫЕ ПРЕДОСТАВЛЯЮТСЯ НА ВХОД СЕТИ

Перед загрузкой в сеть входные данные обрабатываются:

1. Размер изображений изменяется до величины 40x40
2. Значения пикселей преобразуются к вещественному типу данных, приводятся к диапазону [0, 1]

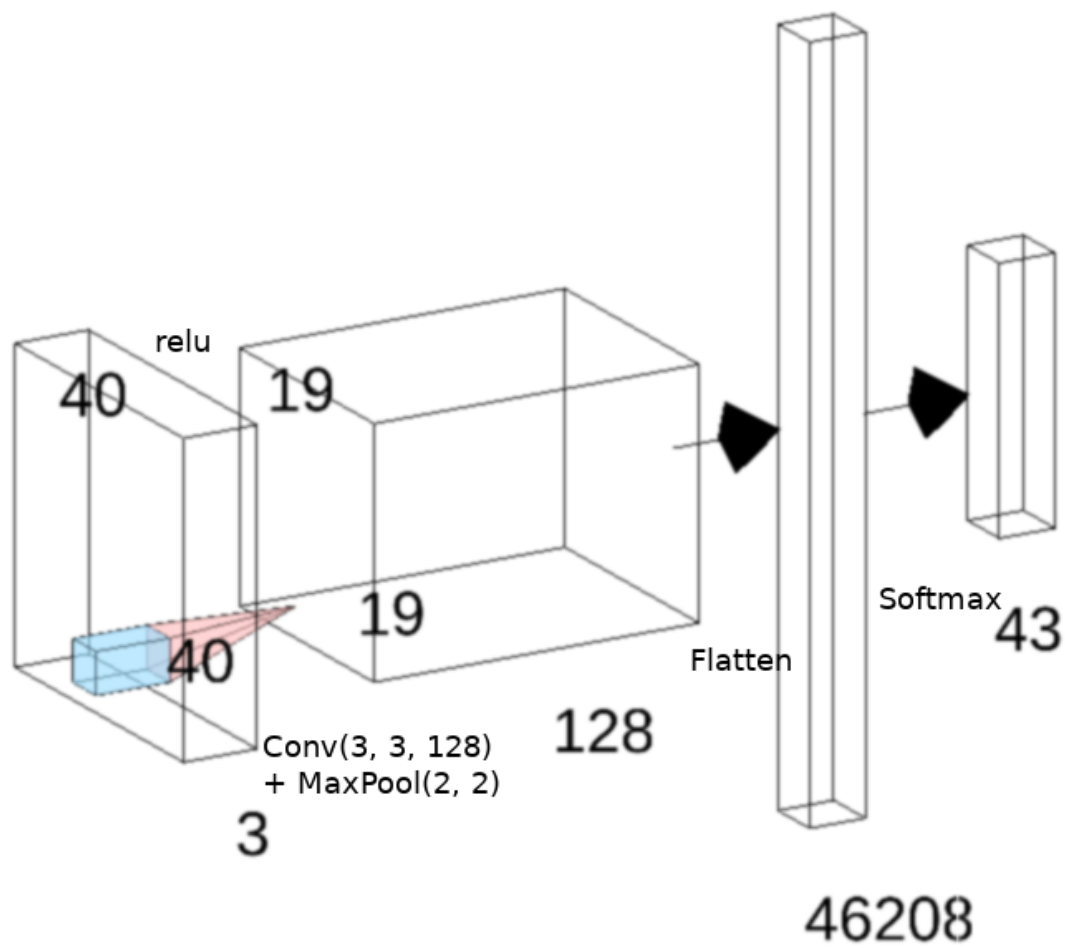
РАЗРАБОТАННЫЕ ПРОГРАММЫ/СКРИПТЫ

Программа состоит из одного файла *Jupyter Notebook*. Код написан на языке программирования *Python*, с помощью *Keras* - оболочки для фреймворка *TensorFlow*.

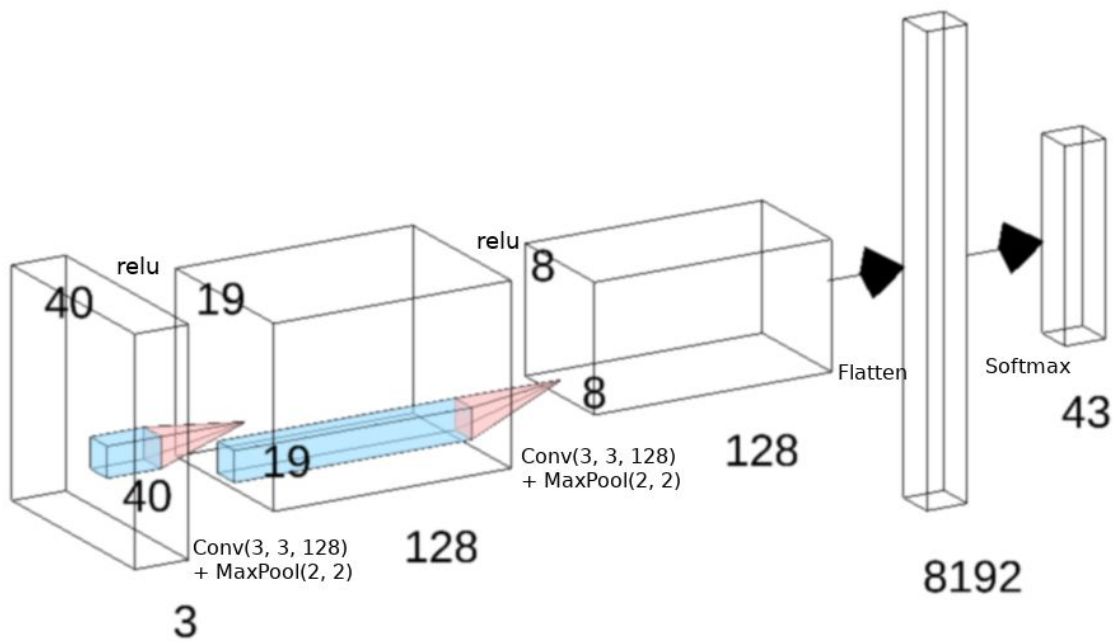
Jupyter Notebook – это крайне удобный инструмент для создания красивых аналитических отчетов, так как он позволяет хранить вместе код, изображения, комментарии, формулы и графики. В *Jupyter Notebook* сразу видно, что возвращает та или иная функция, что особенно важно в начале, при ознакомлении с данными (показывает изображения, статистику и прочее). Выводы дополняются комментариями для более простого понимания кода.

ТЕСТОВЫЕ КОНФИГУРАЦИИ СЕТЕЙ

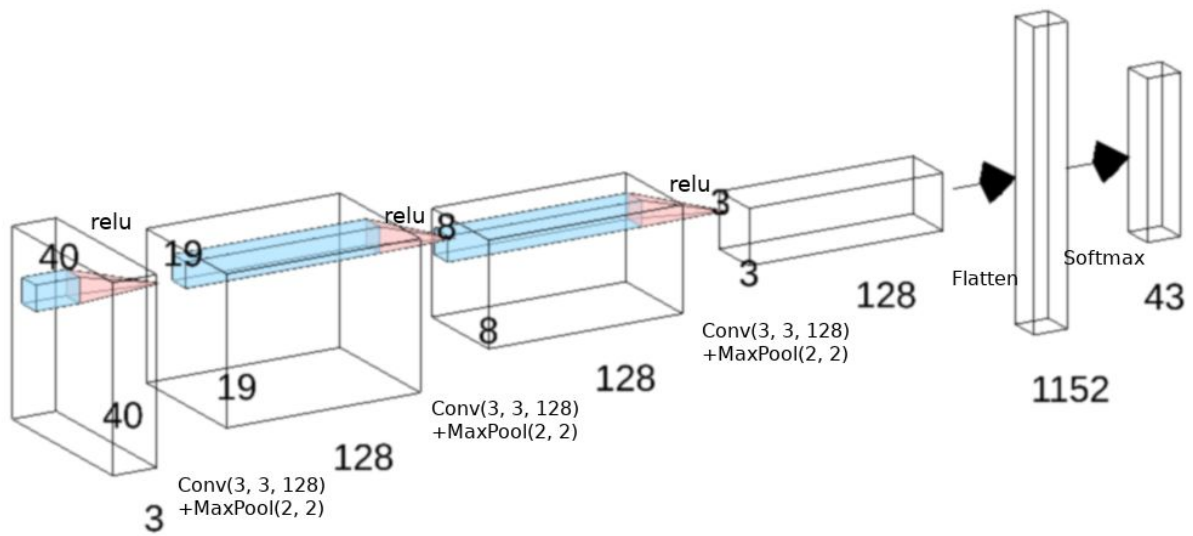
Схемы использованных сетей



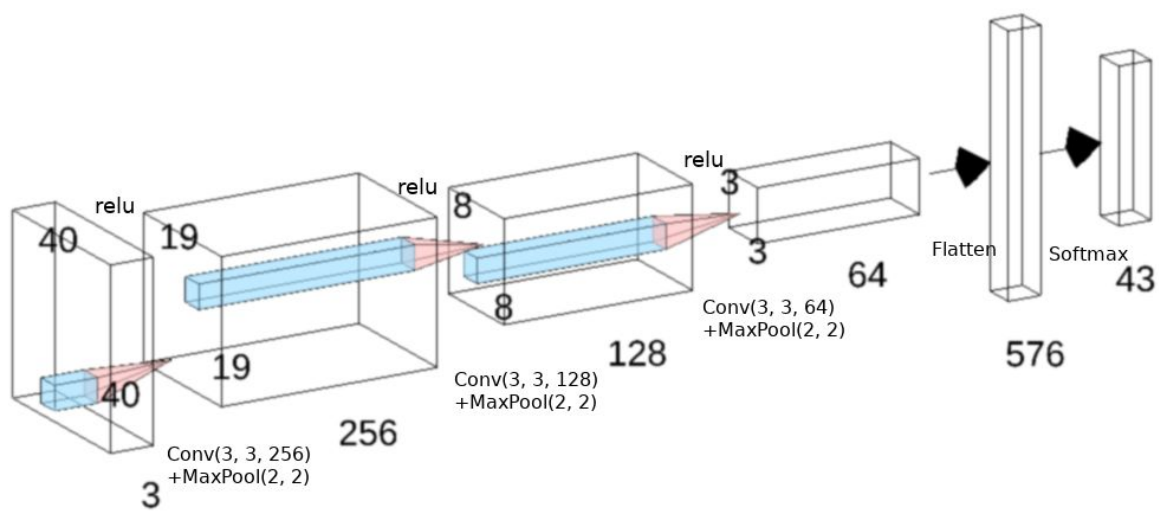
Сеть №1



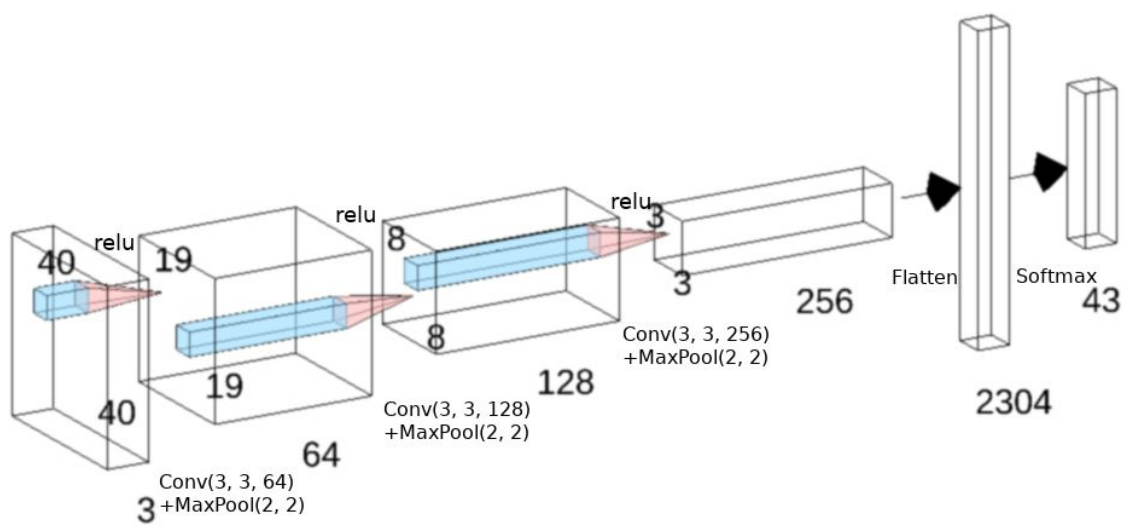
Сеть №2



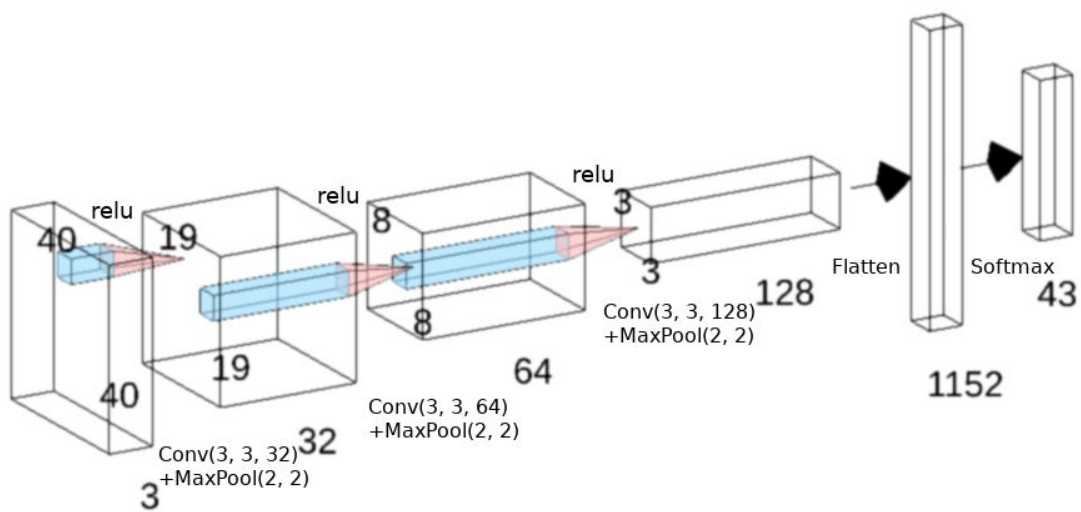
Сеть №3



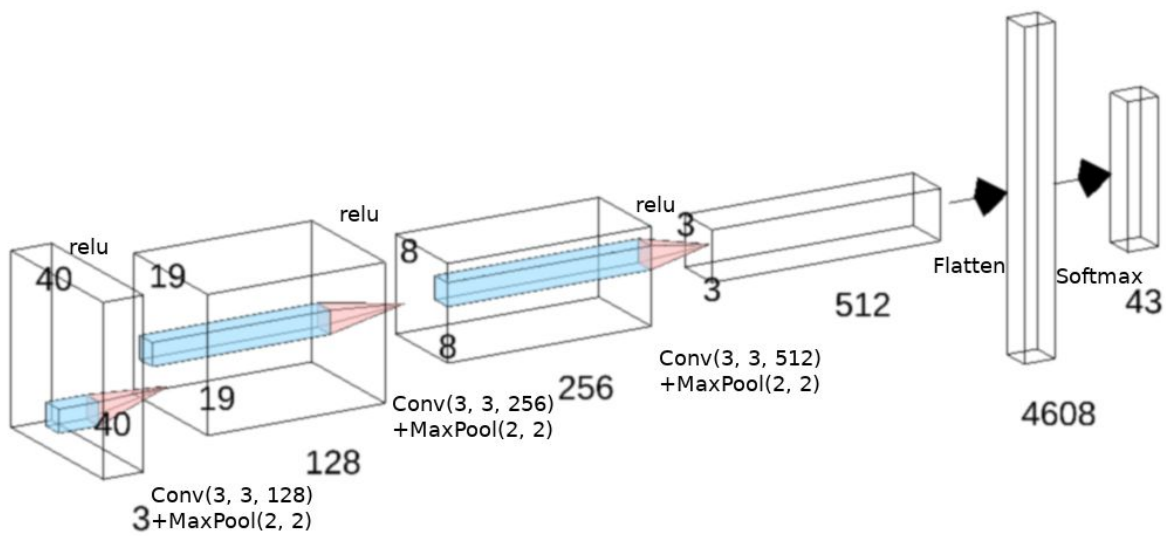
Сеть №4



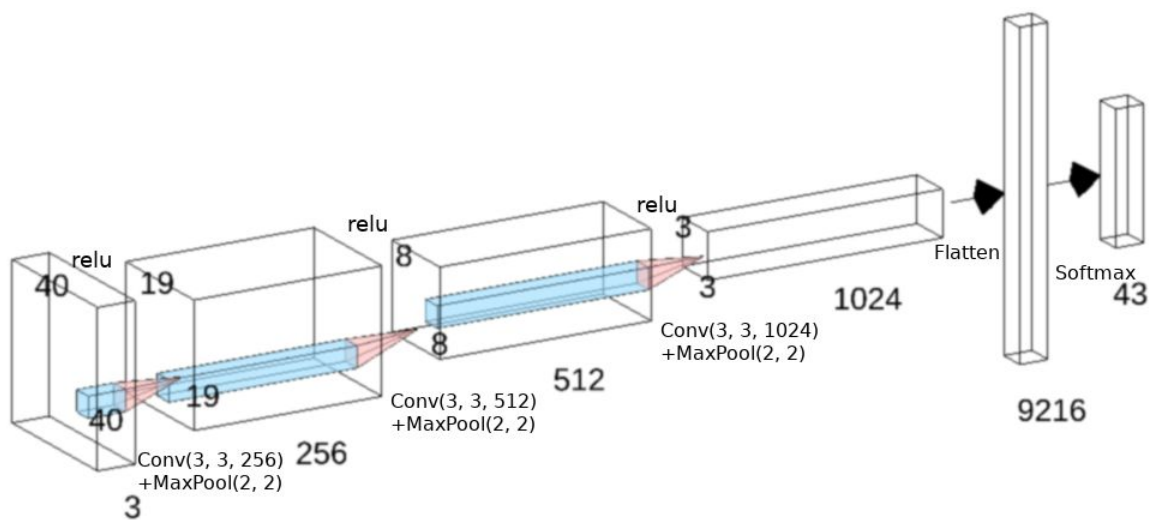
Сеть №5



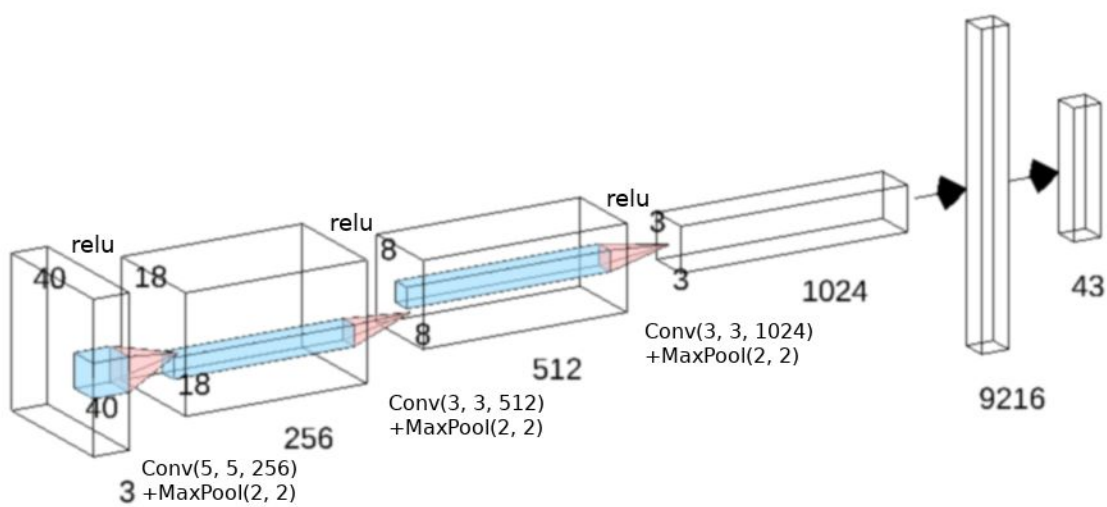
Сеть №6



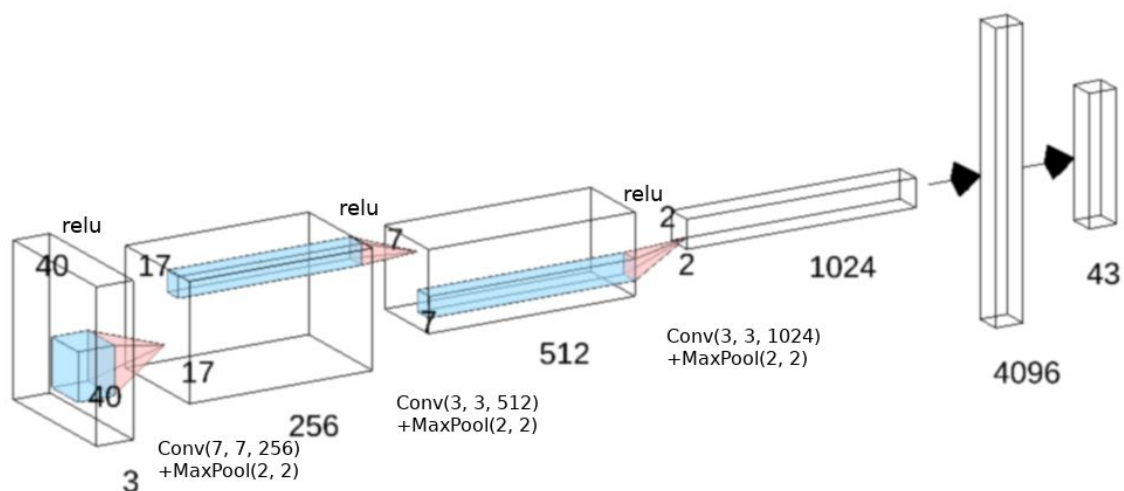
Сеть №7



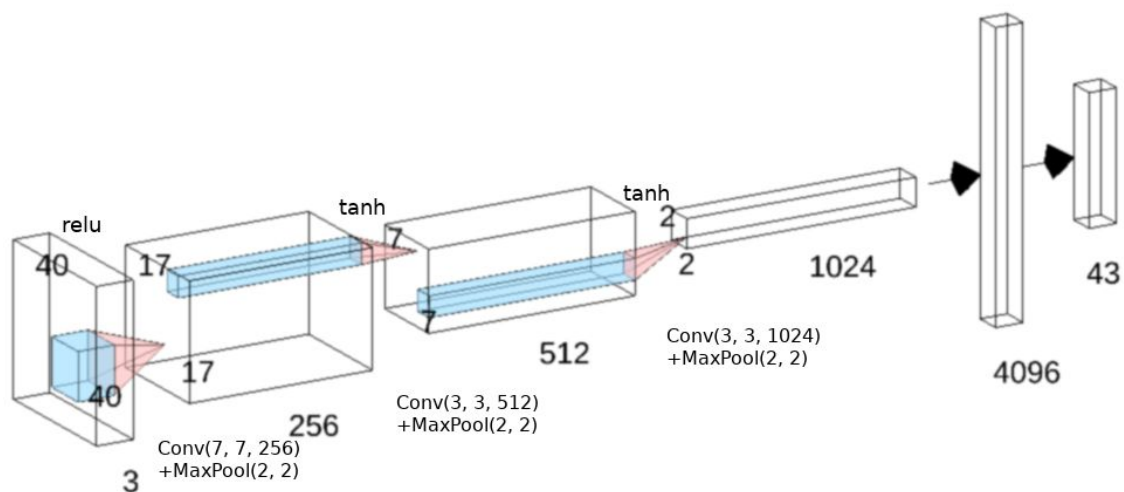
Сеть №8:



Сеть №9:



Сеть №10:



Сеть №11:

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В процессе экспериментов запускалось обучение нейронной сети с различным набором параметров.

Таблица с результатами

Для всех схем нейронных сетей заданы следующие параметры:

Функция оптимизации	Adam
Скорость обучения	0.0005
Количество эпох	20
Функция инициализации	glorot_uniform

№	Слои Conv2D(filters, kernel size, activation function) MaxPooling2D(Pool size)	Тренировочная точность	Тестовая точность	Время обучения (сек)
1	Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2)	99.70%	88.69%	49.2
2	Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2)	99.88%	92.14%	60.6
3	Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(256, (3, 3), relu) MaxPooling2D(2, 2)	99.71%	92.00%	64.8
4	Conv2D(256, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(64, (3, 3), relu) MaxPooling2D(2, 2)	99.72%	91.60%	97.2
5	Conv2D(64, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(256, (3, 3), relu) MaxPooling2D(2, 2)	99.99%	93.10%	53.4
6	Conv2D(32, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(64, (3, 3), relu)	99.64%	90.21%	34.8

	MaxPooling2D(2, 2) Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2)			
7	Conv2D(128, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(256, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(512, (3, 3), relu) MaxPooling2D(2, 2)	99.98%	93.30%	96.6
8	Conv2D(256, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(512, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(1024, (3, 3), relu) MaxPooling2D(2, 2)	100.00%	94.97%	232.8
9	Conv2D(256, (5, 5), relu) MaxPooling2D(2, 2) Conv2D(512, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(1024, (3, 3), relu) MaxPooling2D(2, 2)	100.00%	95.53%	210
10	Conv2D(256, (7, 7), relu) MaxPooling2D(2, 2) Conv2D(512, (3, 3), relu) MaxPooling2D(2, 2) Conv2D(1024, (3, 3), relu) MaxPooling2D(2, 2)	100.00%	95.79%	213
11	Conv2D(256, (7, 7), relu) MaxPooling2D(2, 2) Conv2D(512, (3, 3), tanh) MaxPooling2D(2, 2) Conv2D(1024, (3, 3), tanh) MaxPooling2D(2, 2)	100.00%	95.51%	212.4

Результат, очевидно, хуже по сравнению с валидационной выборкой, но тем не менее, хороший. Причин может быть много. Разница между тестовыми и тренировочными данными может быть большой. Возможно стоит добавлять регуляризацию, Dropout, а также искусственно изменять и добавлять данные в тренировочную выборку (augmentation).