

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ «МИСиС»

ИНСТИТУТ	ИТКН
КАФЕДРА	ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ
НАПРАВЛЕНИЕ	09.04.03 «ПРИКЛАДНАЯ ИНФОРМАТИКА»

Лабораторная работа № 4

по нейронным сетям
«Сверточные нейронные сети»

Группа: МПИ-20-4-2

Студент: Добрынин Владислав

Проверил: Курочкин И.И.

Москва, 2020

Задание:

1. Выбрать/разработать архитектуру сверточной нейронной сети(CNN) для классификации цветных изображений.
2. Решить задачу классификации изображений на эталонном датасете CIFAR-10. Настройку архитектуры CNN и параметров обучения проводить для получения наилучших результатов.
3. Полученные результаты (см. раздел "Результаты") сравнить с опубликованными результатами для аналогичных/похожих архитектур и дать на них ссылку.
4. Используя ту же CNN и полученный набор весов провести дообучить CNN на классах из одного суперкласса эталонного датасета CIFAR-100. Задача классификации решается на классах CIFAR-10 + классы из одного суперкласса CIFAR-100.
5. Провести сравнительный анализ результатов с полученными результатами из п.3

Суперкласс из датасета CIFAR-100: large omnivores and herbivores — camel, cattle, chimpanzee, elephant, kangaroo

Ход выполнения лабораторной работы:

Работа выполнялась на основе исследования Jason Brownlee от 28.08.2020 (<https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/>).

В качестве основного инструмента была выбрана библиотека Keras с использованием backend tensorflow. Полученные нейронные сети для ускорения хода работы обучались при помощи вычислений на графической карте (GPU).

Для каждой сети подсчитывались следующие метрики:

1. Accuracy
2. Loss
3. F1
4. Precision
5. Recall

Результаты

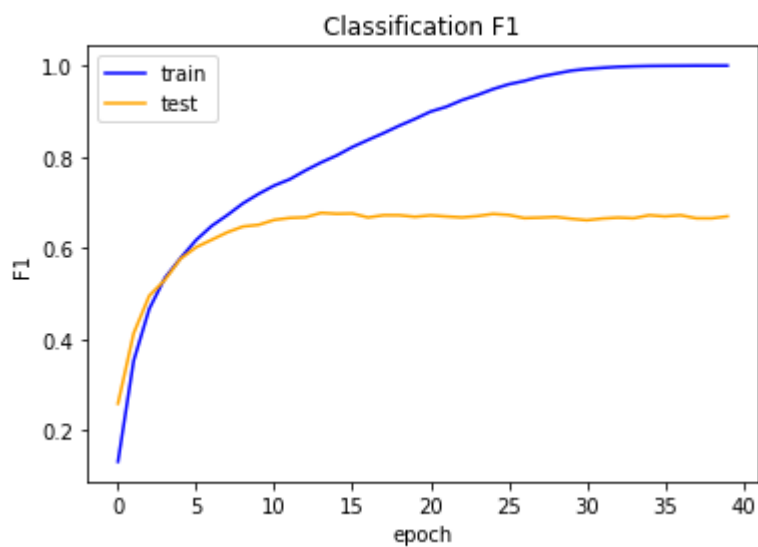
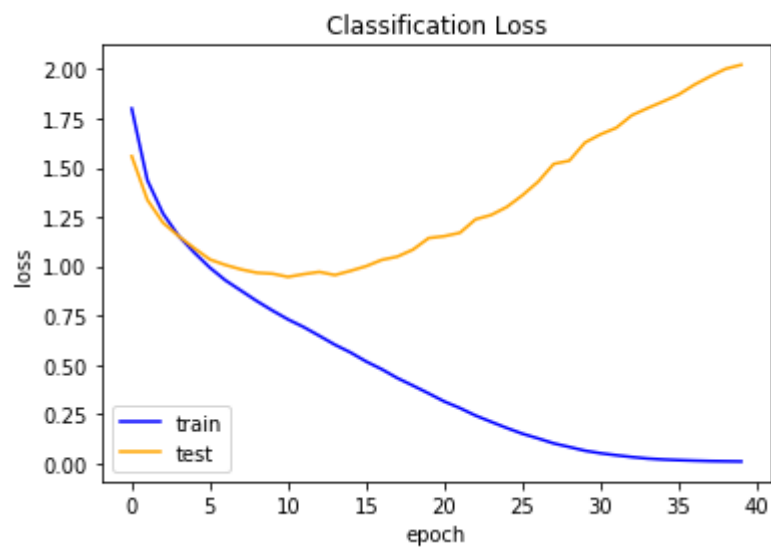
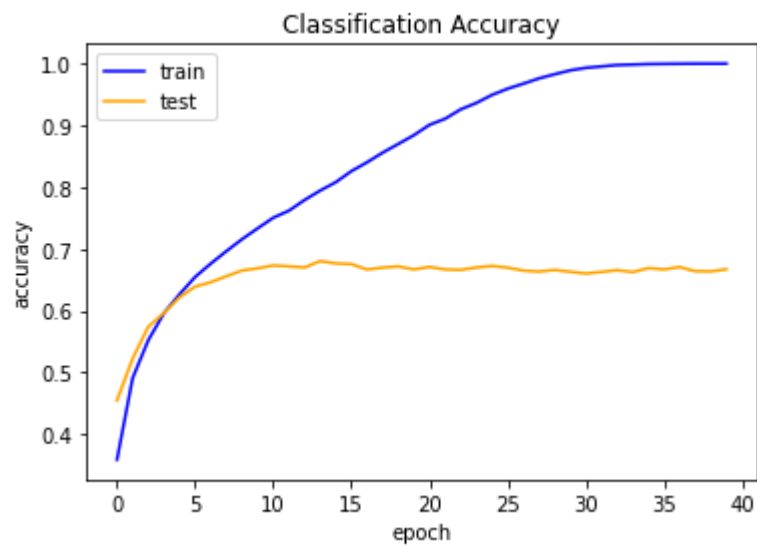
Сеть 1:

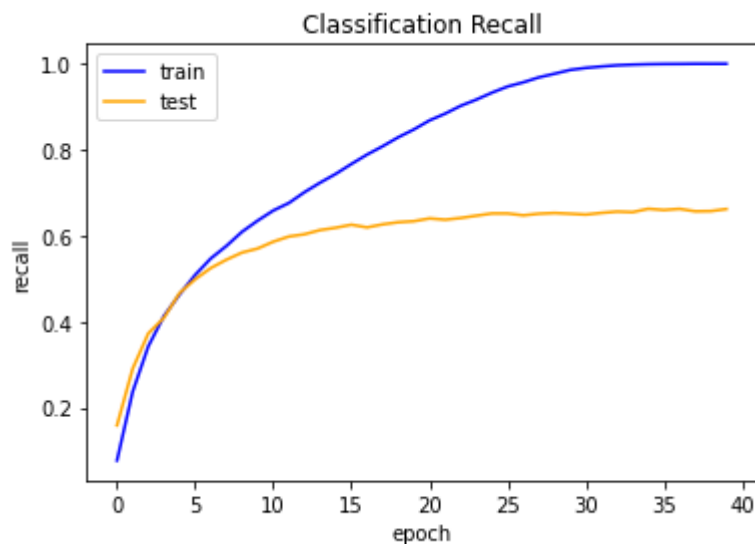
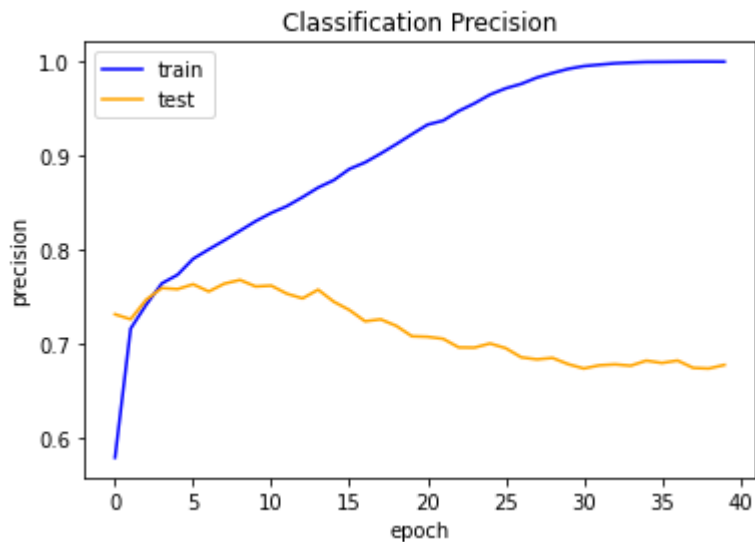
В качестве первой сети было решено использовать аналог архитектуры VGG — несколько сверточных слоев с последующим выбором максимального значения.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 32, 32, 32)	896
conv2d_7 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 32)	0
flatten_3 (Flatten)	(None, 8192)	0
dense_6 (Dense)	(None, 128)	1048704
dense_7 (Dense)	(None, 10)	1290
Total params: 1,060,138		
Trainable params: 1,060,138		
Non-trainable params: 0		

Модель обучалась на датасете CIFAR-10 в течение 40 эпох. Результат представлен в Lab4_Dobrynin_v1.ipynb.

```
accuracy > 66.700  
loss > 202.247  
f1 > 66.917  
precision > 67.700  
recall > 66.174
```





Анализ результатов сети 1:

Как можно видеть, после 15 эпох сеть начинает переобучаться, в итоге падает качество результата. Такая сеть не подходит.

Сеть 2:

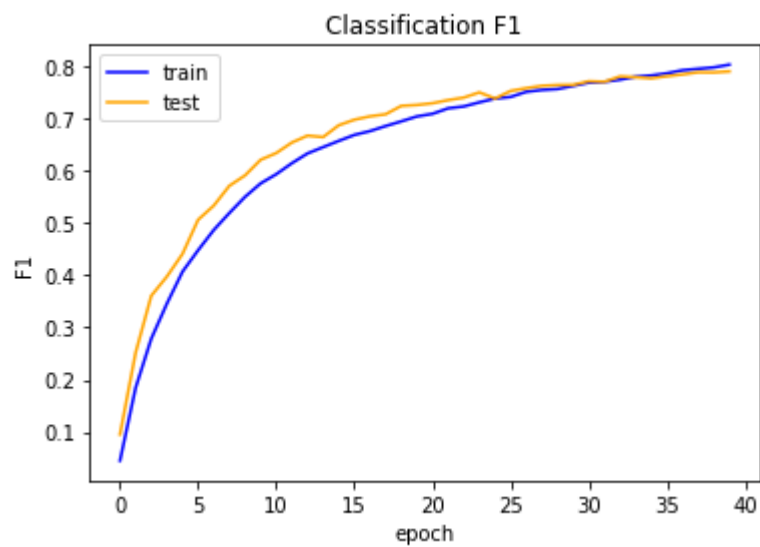
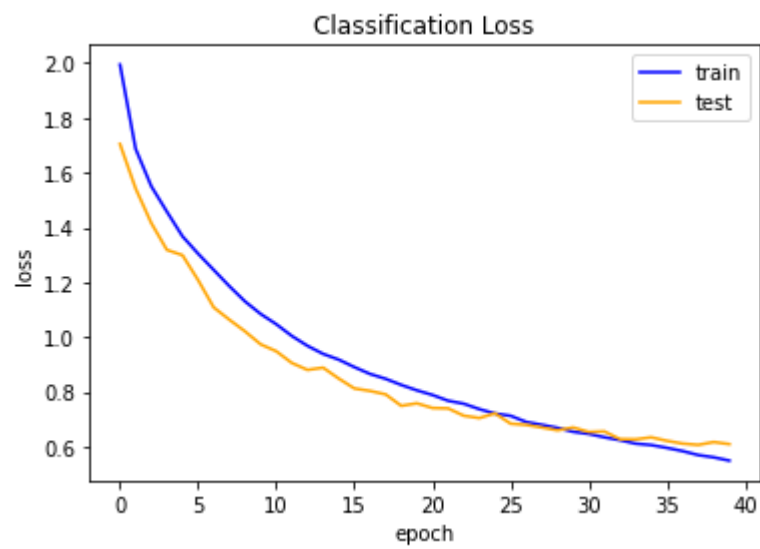
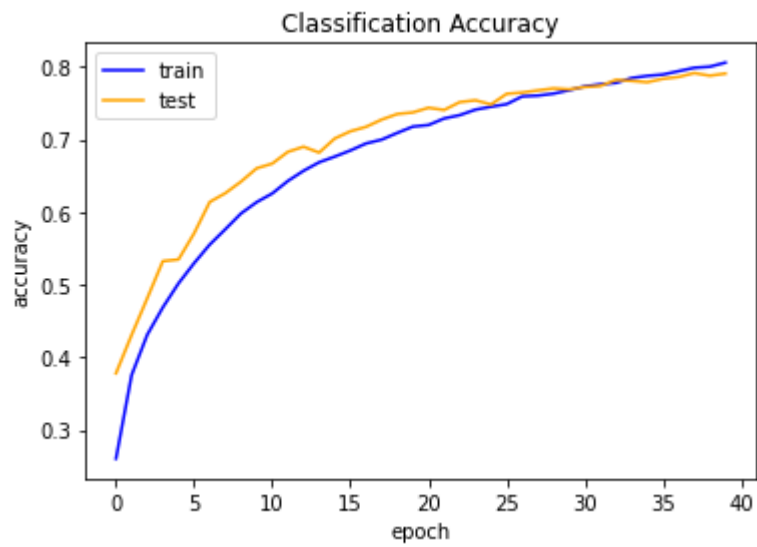
Для модернизации сети из предыдущего пункта было принято решение добавить еще 2 блока архитектуры VGG, а также добавить регуляризацию методом dropout— сеть во время обучения будет случайным образом выключать 20% нейронов. Данный метод должен предотвратить переобучение сети.

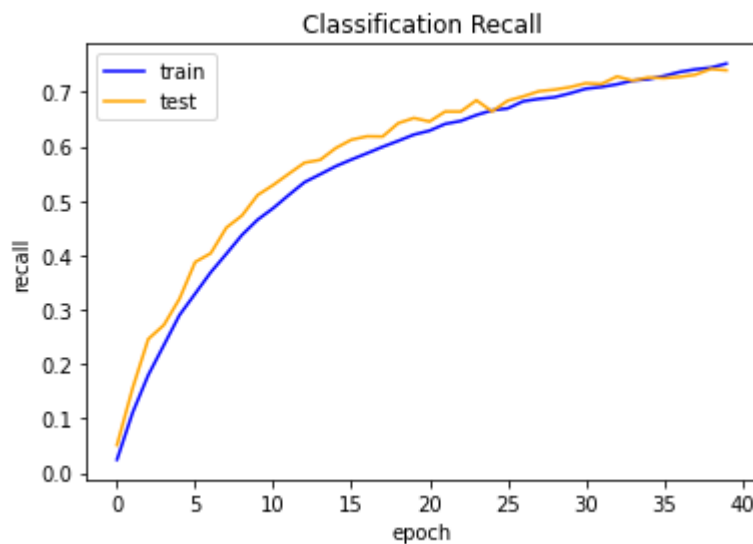
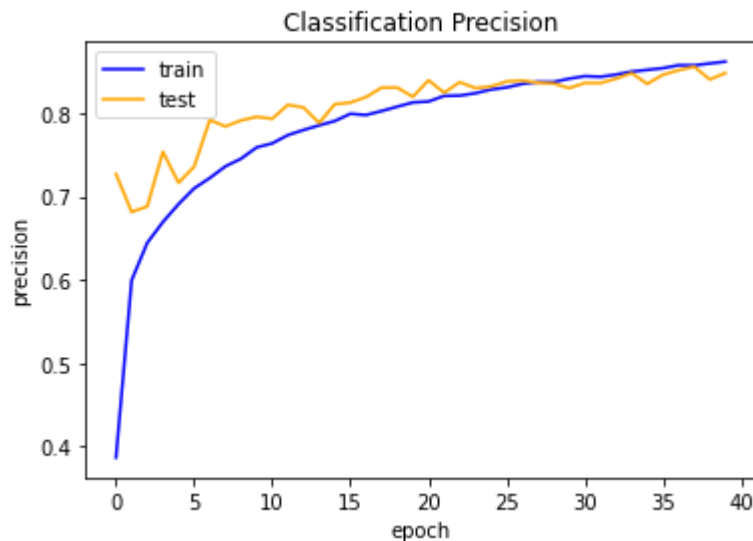
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
conv2d_5 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 128)	262272
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
Total params: 550,570		
Trainable params: 550,570		
Non-trainable params: 0		

Модель обучалась на датасете CIFAR-10 в течение 40 эпох. Результат представлен в Lab4_Dobrynin_v2.ipynb.

Анализ результатов сети 2:

```
accuracy > 79.080
loss > 60.956
f1 > 78.932
precision > 84.809
recall > 74.012
```





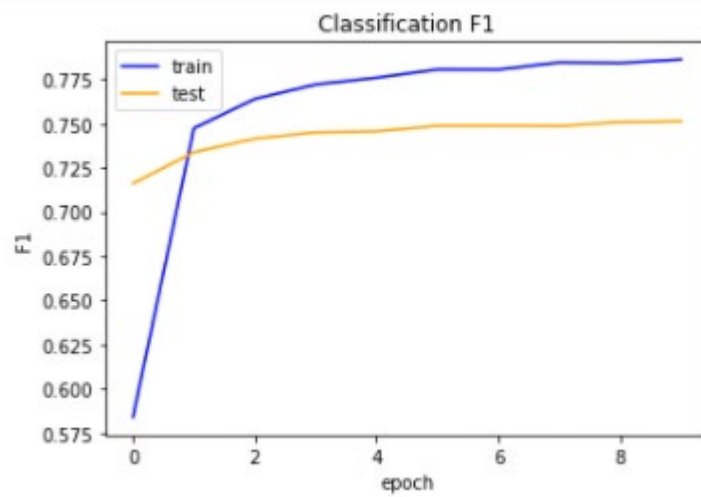
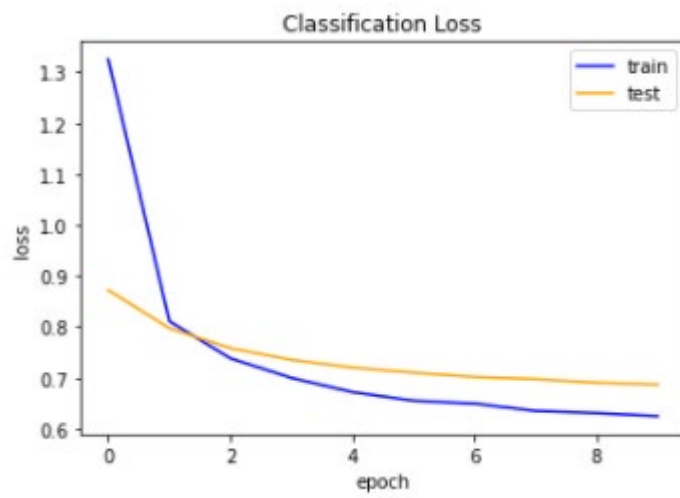
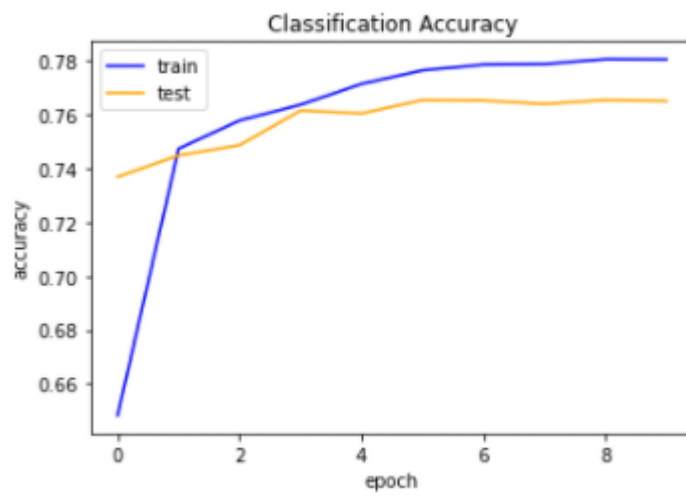
Как можно видеть, сеть работает адекватно и не переобучается. Такая сеть подходит для дальнейшей работы.

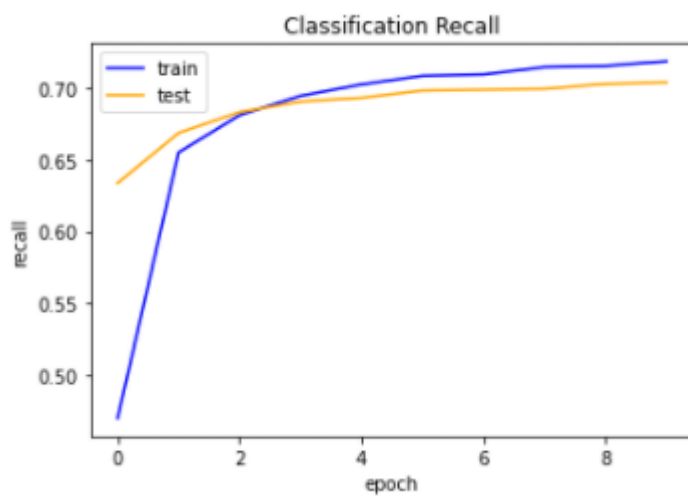
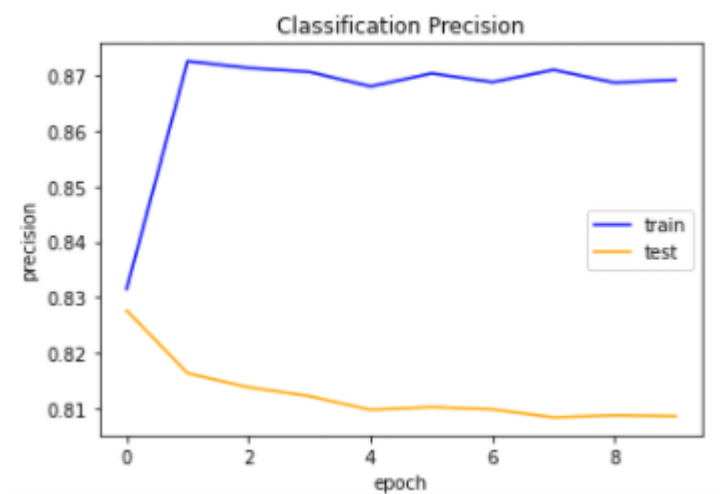
Дальнейшее обучение сети:

Для дальнейшего обучения был выбран суперкласс из библиотеки CIFAR-100 в соответствии с вариантом. Сеть обучалась 10 эпох.

Анализ дальнейшего обучения:

```
accuracy > 76.522  
loss > 68.662  
f1 > 75.288  
precision > 80.831  
recall > 70.640
```



Дообученная сеть показывает хорошие варианты на дополненном наборе данных.

Анализ:

	Сеть 1	Сеть 2 CIFAR10	Сеть 2 с доп. обучением
Accuracy	66.7	79.08	76.5
Loss	202.6	60.9	68.6
F1	66.9	78.9	75.2
Precision	67.7	84.8	80.8
Recall	66.1	74.0	70.6

Сравнение с исходным исследованием

	Сеть 1	Сеть 2 CIFAR10	Сеть 2 с доп. обучением	Исходное исследование CIFAR10
Accuracy	66.7	79.08	76.5	83.5
Loss	202.6	60.9	68.6	~40
F1	66.9	78.9	75.2	No data
Precision	67.7	84.8	80.8	No data
Recall	66.1	74.0	70.6	No data

Несмотря на то, что в исходном исследовании модель обучалась 100 эпох, а в данном исследовании 40 эпох, результат получился схожим.

Вывод:

В результате работы были изучены сверточные нейронные сети. За основу работы было взято исследование Jason Brownlee. Обучение датасета происходило на датасете CIFAR-10. В ходе работы были реализованы две архитектуры нейронных сетей, одна из которых с регуляризацией методом dropout, что помогло избежать переобучения сети. Результат работы можно считать успешным, т.к. были получены результаты, схожие с результатами опорного исследования.