Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

ЛАБОРАТОРНАЯ РАБОТА №6

«Применение сверточных нейронных сетей

(многоклассовая классификация)»

Магистрант: Проверил:

гр. 956241 Заливако С. С.

Шуба И.А.

Минск, 2020

**ХОД РАБОТЫ**

**Задание.**

**Данные:** Набор данных для распознавания языка жестов, который состоит из изображений размерности 28x28 в оттенках серого (значение пикселя от 0 до 255). Каждое из изображений обозначает букву латинского алфавита, обозначенную с помощью жеста, как показано на рисунке ниже (рисунок цветной, а изображения в наборе данных в оттенках серого). Обучающая выборка включает в себя 27,455 изображений, а контрольная выборка содержит 7172 изображения. Данные в виде csv-файлов можно скачать на сайте Kaggle -> <https://www.kaggle.com/datamunge/sign-language-mnist>

**Задание 1.**

Загрузите данные. Разделите исходный набор данных на обучающую и валидационную выборки.

**Задание 2.**

Реализуйте глубокую нейронную сеть со сверточными слоями. Какое качество классификации получено? Какая архитектура сети была использована?

**Задание 3.**

Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?

**Задание 4.**

Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Можно ли было обойтись без него?

Какой максимальный результат удалось получить на контрольной выборке?

**Результат выполнения:**

**Задание 1.** Загрузите данные. Разделите исходный набор данных на обучающую и валидационную выборки.

Объявим несколько констант и загрузим наборы данных.





Рисунок 1 – Загрузка данных

Пример датафрейма.

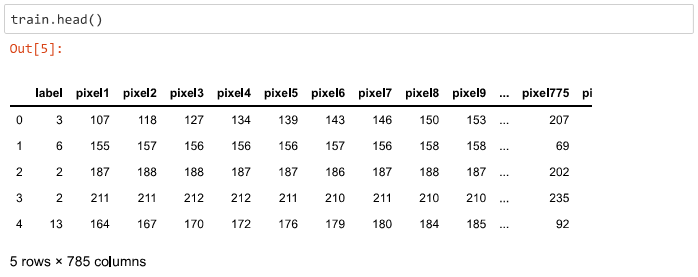


Рисунок 2 – Пример датафрейма

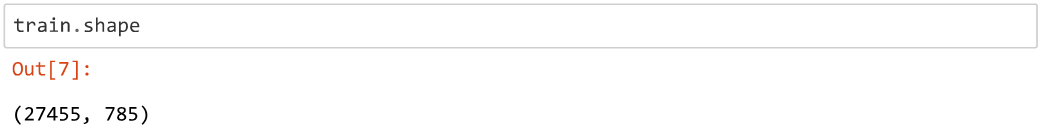
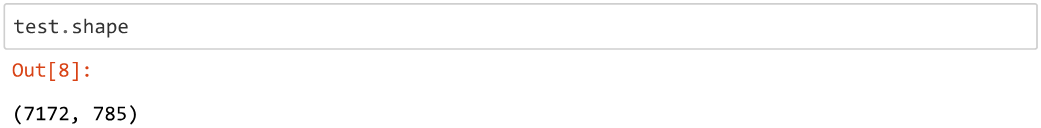
 

Рисунок 3 – Размеры наборов данных



Рисунок 4 – Получение лейблов и удаление колонки label

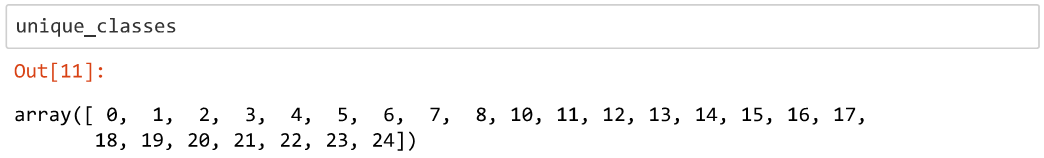


Рисунок 5 – Количество уникальных классов

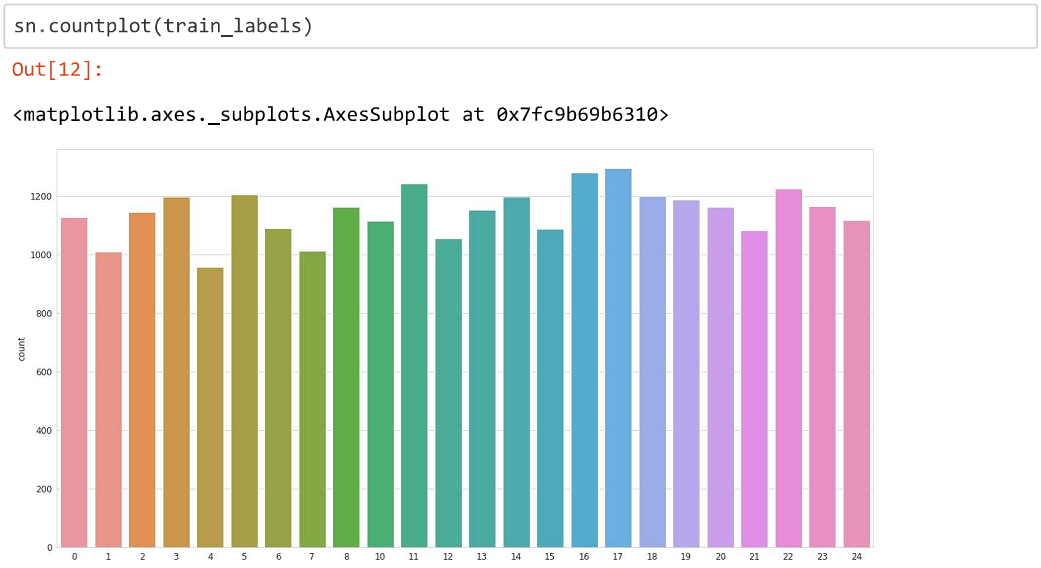


Рисунок 6 – Проверка сбалансированности классов

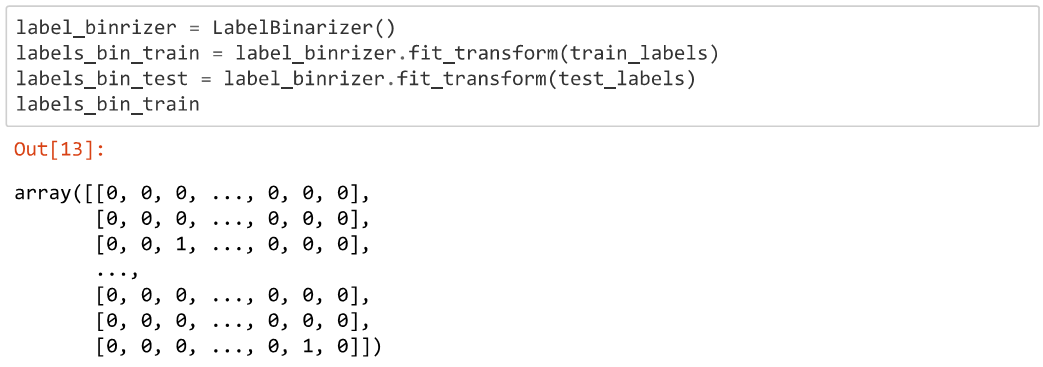




Рисунок 7 – Бинаризация лейблов и нормализация изображений

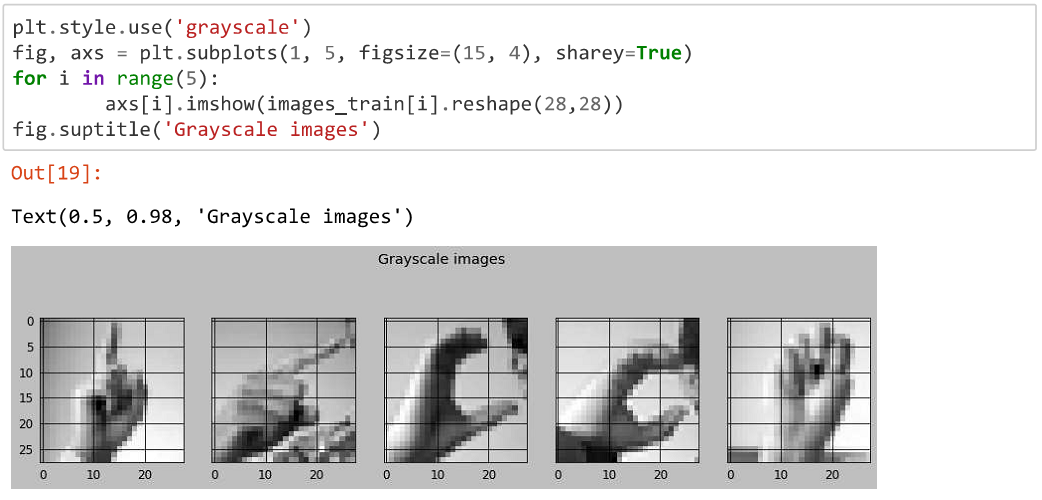
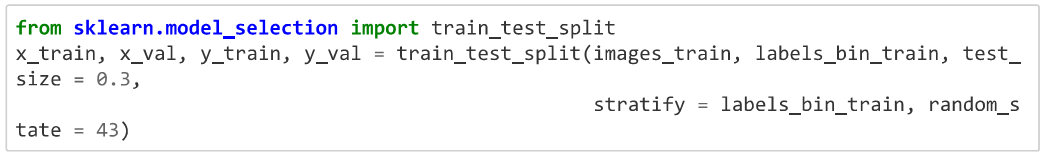


Рисунок 8 – Примеры изображений



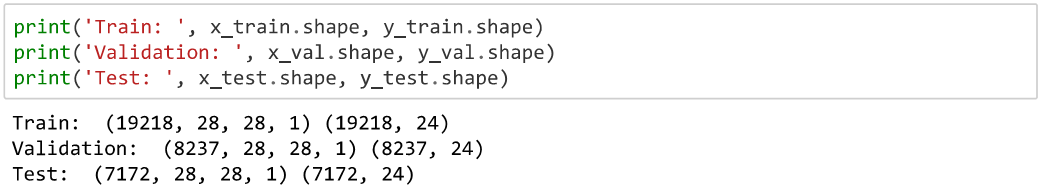
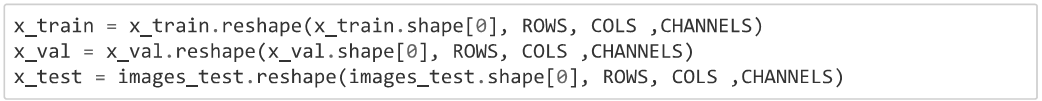


Рисунок 9 – Разделение тренировочных данных и изменение

формы вектора фич

**Задание 2.** Реализуйте глубокую нейронную сеть со сверточными слоями. Какое качество классификации получено? Какая архитектура сети была использована?

Реализуем глубокую сверточную сеть.

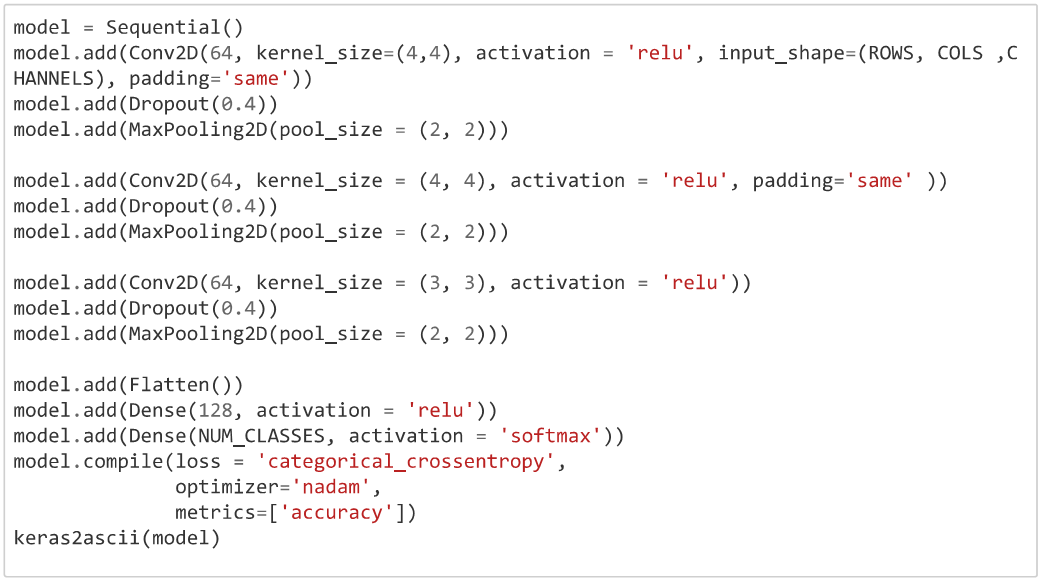


Рисунок 10 – Глубокая сверточная нейронная сеть

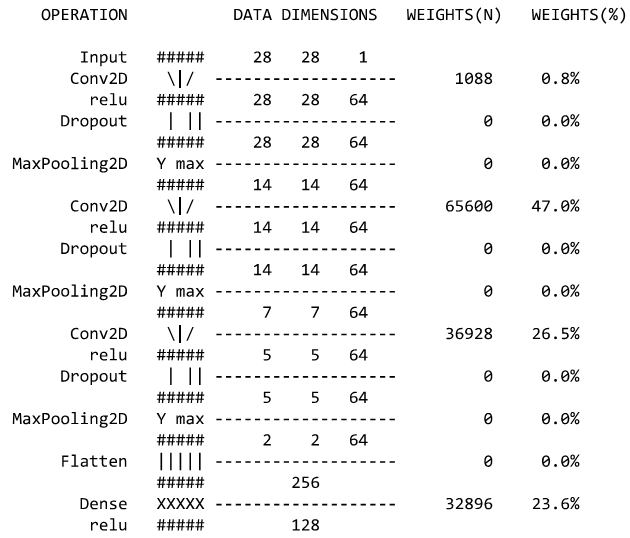




Рисунок 11 – Схема глубокой сверточной нейронной сети

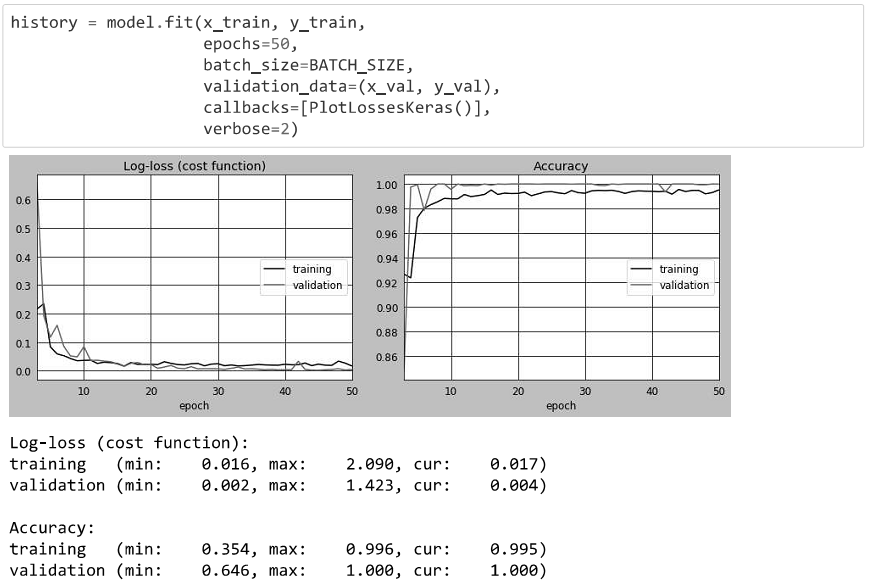


Рисунок 12 – Результаты обучения

Как видно из графиков, точность на валидационных данных приблизительно такая же, как на тренировочных. Максимальная точность на валидационных данных составила 1, а минимальный log-loss – 0,002.

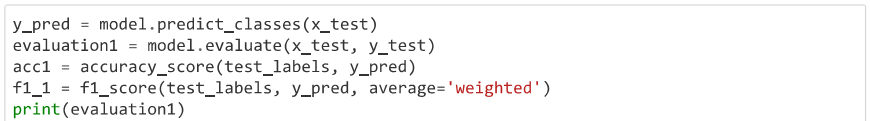
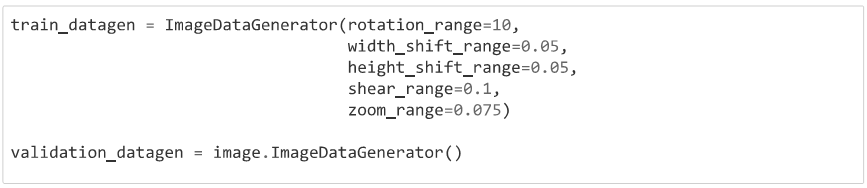


Рисунок 13 – Результаты модели на контрольной выборке

Таким образом, точность модели составила 0,97, а log-loss 0.099.

**Задание 3.** Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?

Применим дополнение данных путем генерации изображений с некоторыми преобразованиями.



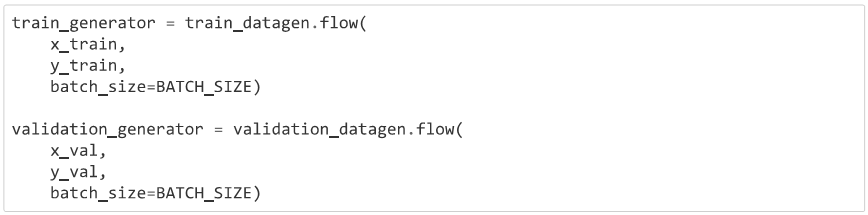


Рисунок 14 – Генерация изображений с некоторыми преобразованиями

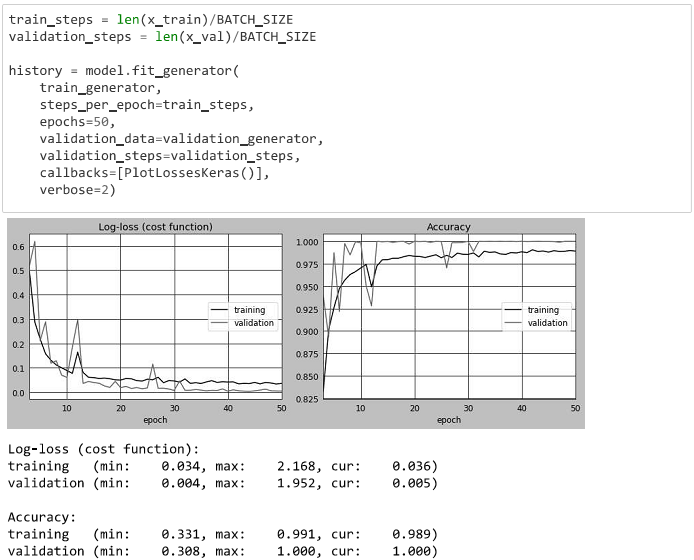


Рисунок 15 – Результаты обучения

Как видно из графиков, точность и log-loss на валидационных данных приблизительно такая же как и на, чем на тренировочных. Максимальная точность на валидационных данных составила 1, а минимальный log-loss – 0,004.

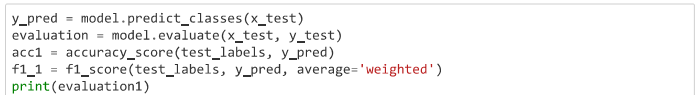


Рисунок 16 – Результаты модели на контрольной выборке

Таким образом, точность модели составила 0,99, а log-loss 0,019.

**Задание 4.** Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Можно ли было обойтись без него? Какой максимальный результат удалось получить на контрольной выборке?

Чтобы применить передаточное обучение, необходимо подстроить форму данных, подаваемых на вход предобученной сети.

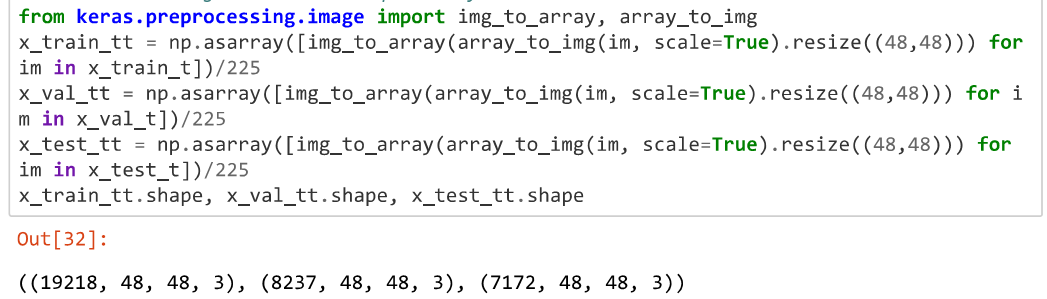
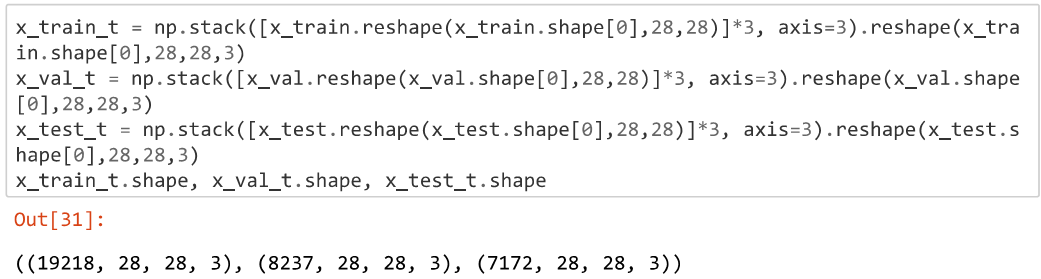


Рисунок 17 –Настройка входных данных под вход предобученной

сети VGG16

Поэксперементируем с сетью VGG16.



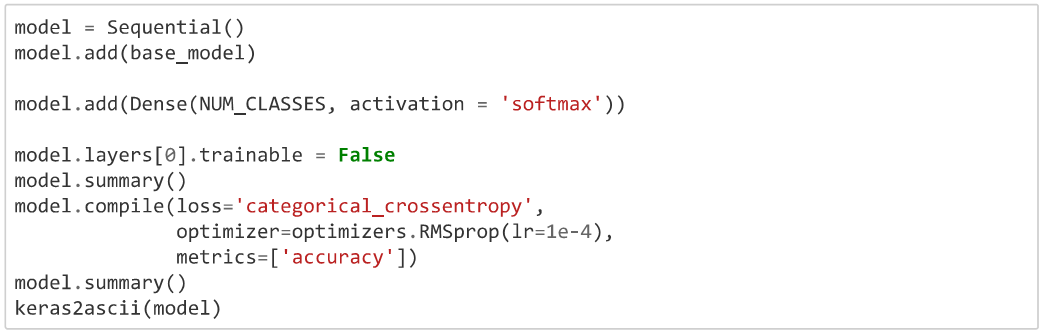


Рисунок 18 –Нейронная сеть с предобученной сетью VGG16

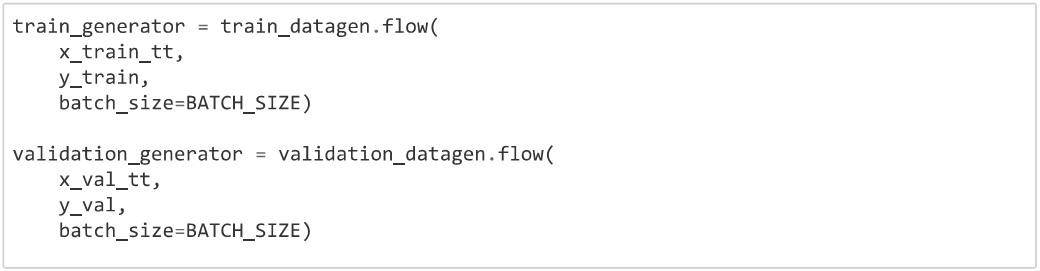


Рисунок 19 – Генерация изображений с некоторыми преобразованиями

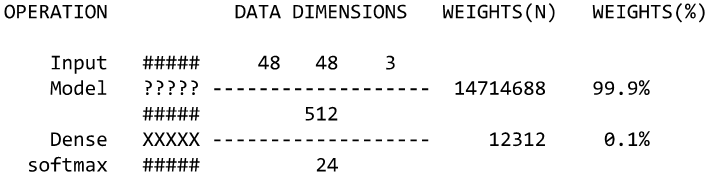


Рисунок 20 – Схема нейронной сети с предобученной сетью VGG16

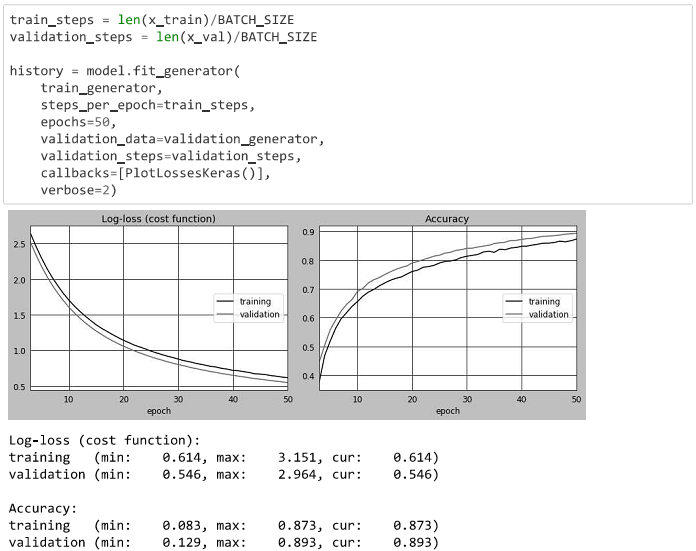


Рисунок 21 – Результаты обучения

Как видно из графиков, точность и log-loss на валидационных данных приблизительно такая же, как на тренировочных. Максимальная точность на валидационных данных составила 0,893, а минимальный log-loss – 0,546.

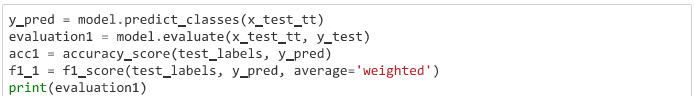
 

Рисунок 22 – Результаты модели на контрольной выборке

Таким образом, точность модели составила 0,8, а log-loss 0.73 при обучении на 50 эпохах. Из графиков точности и log-loss заметим, что при увеличении количества эпох скор будет увеличиваться. При обучении с 50 эпохами, общая точность модели осталась достаточно низкой, однако виден положительный тренд – при увеличении количества эпох при обучении, точность будет повышаться, однако достаточно медленно, нежели с применением реализованных выше архитектур. Качество классификатора заметно ниже, нежели обученной в прошлом задании, поэтому в данной задаче можно было бы обойтись без предобученной модели.

**Вывод:**

В ходе выполнения лабораторной работы был изучен датасет с жестами, были реализованы сверточные сети с разными архитектурами, максимальный скор, который удалось получить, составил 0,99, а минимальный log-loss – 0,019. Причем в модели не наблюдалось явного переобучения, так как графики точности на тренировочных и валидационных данных практически повторяли друг друга с ростом числа эпох. Было реализовано передаточное обучение. В качестве предобученной модели была использована одна из популярных сетей VGG16. Однако точность реализованной была хуже, чем созданной ранее, так точность составила 0,8.