Министерство образования и науки Российской Федерации

Южно-Уральский государственный университет

Кафедра системного программирования

**ОТЧЕТ**

**о производственной (преддипломной) практике**

Выполнил:

В.А. Петров, группа КЭ-401

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики:

Ст. преподаватель Федянина Р.С.

В качестве первого листа данного отчета должно быть приложено задание по производственной (преддипломной) практике, подписанное руководителем практики, студентом и утвержденное заведующим кафедрой.

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 3](#_Toc515523785)

[1. ИСХОДНЫЕ ДАННЫЕ 4](#_Toc515523786)

[2. ПОСТАНОВКА ЗАДАЧИ 5](#_Toc515523787)

[3. РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ 5](#_Toc515523788)

[4. ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ 6](#_Toc515523789)

[5. ТЕСТИРОВАНИЕ 8](#_Toc515523790)

[5.1. Тестирование нейронной сети 8](#_Toc515523791)

[5.2. Тестирование настольного приложения 9](#_Toc515523792)

[РЕЗУЛЬТАТЫ 11](#_Toc515523793)

# ИСХОДНЫЕ ДАННЫЕ

Основной целью моей выпускной квалификационной работы является разработать настольное приложение, которое позволяет по видеофрагменту с записью мимики определить психологический подтип человека (этик, логик) в классификации К.Г. Юнга.

На момент выхода на преддипломную практику были завершены следующие этапы:

1. Анализ предметной области;
2. Теоретическая часть;
3. Проектирование;
4. Реализация предобработки видеофрагмента и приложения.

На первом этапе мной был осуществлен обзор существующих подходов к программному анализу мимики человека, которые могут быть использованы для определения психологического подтипа человека. На данном этапе было принято решение применить нейросетевой подход, как наиболее перспективный.

На втором этапе мной были рассмотрены основные этапы предварительной обработки видеофрагмента, а также рассмотрены теоритические сведения об искусственных нейронных сетях и их разновидностях топологий.

На третьем этапе был спроектирован алгоритм предварительной обработки видеофрагмента, топология искусственной нейронной сети, архитектура приложения. Описаны функциональные и нефункциональные требования к приложению.

На четвертом этапе была осуществлена программная реализация алгоритма предобработки видеофрагмента и настольного приложения.

# ПОСТАНОВКА ЗАДАЧИ

В ходе преддипломной практики необходимо было решить следующие задачи:

1. Реализация спроектированной архитектуры нейронной сети;
2. Обучение и тестирование нейронной сети;
3. Тестирование настольного приложения.

# РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ

Для решения задачи ВКР на этапе проектирования была разработана топология нейронной сети, представленная на рисунке 1.

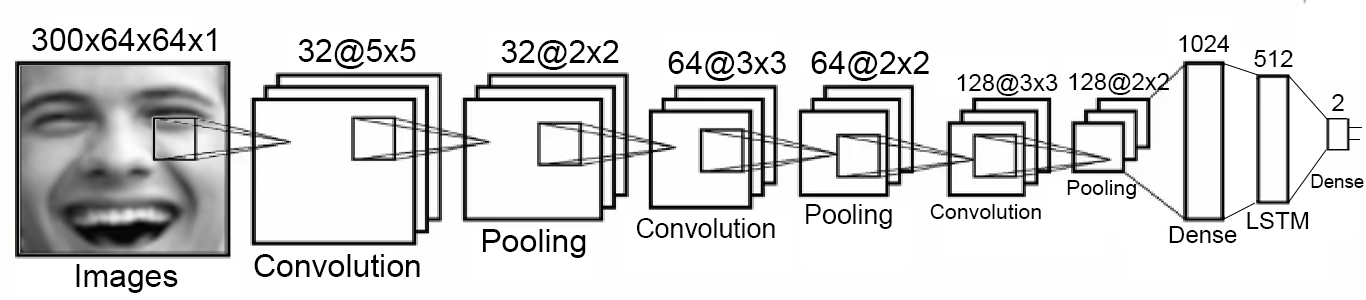


Рис. 1. Топология нейронной сети

Для реализации данной топологии искусственной нейронной сети мной был выбран язык программирования Python 3 и программные библиотеки Keras, Tensorflow. Процесс написания программного кода и его отладки осуществлялась в среде разработки PyCharm Community Edition 2017.1. Также использовалась система контроля версий Git, а в качестве удаленного репозитория был выбран веб-сервис Bitbucket. В листинге 1 представлен программный код реализованной топологии.

Листинг 1. Программный код топологии нейронной сети

|  |
| --- |
| model = Sequential()  model.add(TimeDistributed(Conv2D(32, (5, 5), padding='same', activation='relu'), input\_shape=(None, 64, 64, 1)))  model.add(TimeDistributed(MaxPooling2D(pool\_size=(2, 2))))  model.add(Dropout(0.2))  model.add(TimeDistributed(Conv2D(64, (3, 3), padding='same', activation='relu')))  model.add(TimeDistributed(MaxPooling2D(pool\_size=(2, 2))))  model.add(Dropout(0.2))  model.add(TimeDistributed(Conv2D(128, (3, 3), padding='same', activation='relu')))  model.add(TimeDistributed(MaxPooling2D(pool\_size=(2, 2))))  model.add(Dropout(0.2))  model.add(TimeDistributed(Flatten()))  model.add(TimeDistributed(Dense(1024, activation='relu')))  model.add(Dropout(0.25))  model.add(LSTM(512, dropout=0.1, return\_sequences=False))  model.add(Dense(number\_class, activation='softmax')) |

Данная нейронная сеть состоит из трех слоев свертки, трех слоев субдискритезации, одного рекуррентного слоя и двух полносвязных слоев. На вход нейронная сеть ожидает 4 мерный массив, размером Nx64x64x1, где N – количество кадров в видеофайле. На выходе нейронная сеть выдает массив вероятностей, в котором каждая ячейка содержится численное значение вероятности, что человек, запечатленный на видеофрагменте, принадлежит к заданному психологическому подтипу.

# ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Для обучения и тестирования нейронной сети была подготовлена выборка, состоящая из 262 видеофайлов. На данных видеофайлах записана мимика ярких представителей классифицируемых психологических подтипов. Все видеофайлы приведены к одному формату: длительность – 10 секунд, частота – 30 кадров в секунду, размер – 1280х720 пикселей, формат данных – mp4.

Для обучения нейронной сети было задействовано 80 % от общей выборки, что составило объем в 209 видеофайлов. Ошибка рассчитывалась по функции categorical crossentropy:

где: – класс, представляющий заданный психологический подтип;

– «истинная» вероятность попадания объекта в класс ;

– вероятность, полученная на выходе нейронной сети.

В листинге 2 представлен программный код для запуска процесса обучения нейронной сети.

Листинг 2. Программный код для запуска процесса обучения

|  |
| --- |
| train\_data, train\_answer, test\_data, test\_answer = \  preprocessing.get\_dataset(options.PATH\_TO\_DATA, options.IMAGE\_SIZE,  options.TO\_TRAIN, options.NUMBER\_CLASS)  model = model\_4.get\_model(options.IMAGE\_SIZE, options.NUMBER\_CLASS)  model.compile(loss=keras.losses.categorical\_crossentropy,  optimizer=keras.optimizers.Adadelta(),  metrics=['categorical\_accuracy'])  model.fit(train\_data, train\_answer, batch\_size=options.BATCH\_SIZE,  epochs=options.NUMBER\_EPOCH,  validation\_data=(test\_data, test\_answer), shuffle=True)  model\_4.save\_model(model) |

Обучение нейронной сети проводилось на GPU NVIDIA GeForce GTX 1080 и производилось в течение 20 эпох. В среднем на одну эпоху тратилось 82,7 секунды. Протокол обучения нейронной сети представлен на рисунке 2.

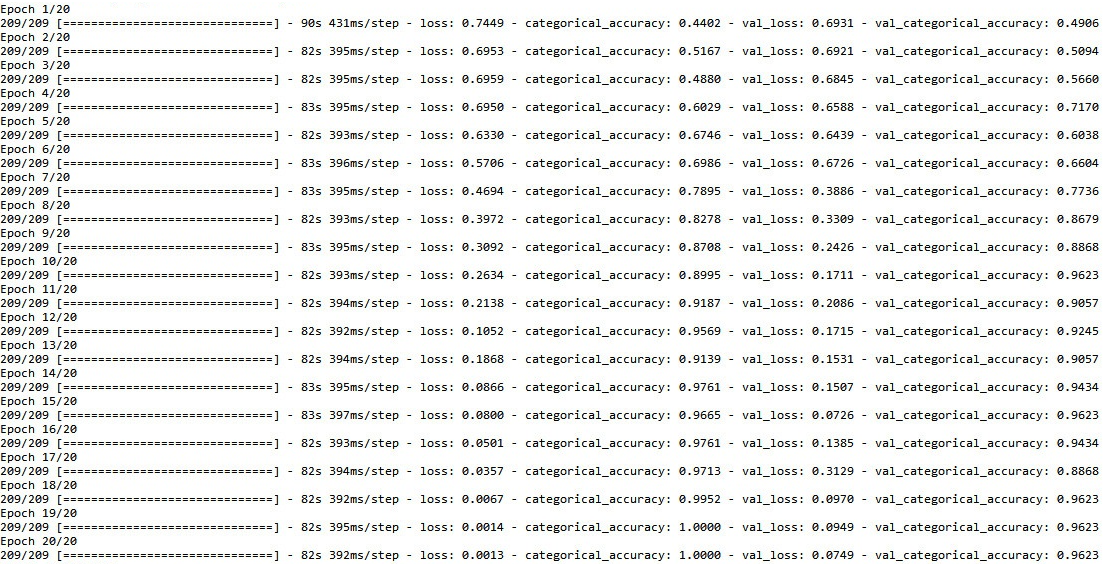


Рис. 2. Проток обучения нейронной сети

# ТЕСТИРОВАНИЕ

# Тестирование нейронной сети

Процесс тестирования нейронной сети заключается в оценки качества обученной модели. Для оценки качества нейронной сети была использована тестовая выборка, которая имела объем в 20 % от общей выборки, что составило 53 видеофайла. В тестовую выборку вошли все видеофайлы, которые не попали в обучающую выборку.

Оценка качества производилась по формуле categorical accuracy:

где – количество правильно классифицированных элементов тестовой выборки, а – количество элементов тестовой выборки.

По итогам тестирования обученная модель показала качество в 96,23 %. Это означало, что нейронная сеть ошиблась в двух элементах тестовой выборки. Был произведен анализ возможной причины ошибки на этих видеофайлах.

В первом неправильно классифицируемом видеофайле, кадр из которого представлен на рисунке 3, запечатлен представитель негроидной расы, являющийся единственным представителем своей расы в выборке.



Рис. 3. Кадр из первого неправильно классифицируемого видеофайла

Это позволило сделать предположение, что из-за того, что это единственный представитель негроидной расы у нейронной сети не было возможности обучиться правильно выделять признаковое описание мимики у такого типа людей.

На втором видеофайле, кадр из которого представлен на рисунке 4, человек находится далеко в кадре и из-за этого нейронная сеть не смогла получить достаточное признаковое описание мимики человека.



Рис. 4. Кадр из второго неправильно классифицируемого видеофайла

# Тестирование настольного приложения

Для проверки на соответствие разработанного настольного приложения функциональным требованиям было проведено функциональное тестирование приложения. Для этого был подготовлен набор тестов, который покрывает функциональные требования к приложению, составленных на этапе проектирования. Затем было проведено тестирование, протокол которого представлен ниже.

Тест №1. Цель: Проверить возможность выбора видеофайла.

Действие: Пользователь нажимает на кнопку «Выбрать видеофайл».

Ожидаемый результат: Пользователь увидит окно выборы видеофайла.

Тест пройден? Да.

Тест №2. Цель: Проверить возможность загрузки видеофайла.

Действие: Пользователь нажимает на кнопку «Выбрать видеофайл», выбирает видеофайл и нажимает кнопку «Открыть».

Ожидаемый результат: Видеофайл загрузился, и пользователь увидел уведомление.

Тест пройден? Да.

Тест №3. Цель: Проверить возможность классификации психологического подтипа человека.

Действие: Пользователь нажимает кнопку «Определить психотип» с выбранным видеофайлом.

Ожидаемый результат: Пользователь увидит сообщение о запуске процесса определения психотипа. В процессе пользователь будет получать информацию о текущем прогрессе. И в конце, пользователь получит результат.

Тест пройден? Да.

Тест №4. Цель: Проверить возможность прервать процесс классификации психологического подтипа человека.

Действие: Пользователь нажимает кнопку «Определить психотип» с выбранным видеофайлом, после того как начался процесс классификации, пользователь нажимает кнопку «Отмена».

Ожидаемый результат: Процесс классификации прервется, и пользователь увидит об этом сообщение.

Тест пройден? Да.

Тест №5. Цель: Проверить возможность выбора модели нейронной сети.

Действие: Пользователь нажимает на кнопку со списком моделей.

Ожидаемый результат: Пользователь увидит список доступных моделей и сможет выбрать любую.

Тест пройден? Да.

Все тесты успешно пройдены.

# РЕЗУЛЬТАТЫ

В рамках прохождения преддипломной практики была успешно разработана нейронная сеть для системы распознавания психологического подтипа человека по классификации К.Г. Юнга. В ходе разработки были решены следующие задачи:

1. Реализована спроектирована архитектура нейронной сети;
2. Обучена и протестирована нейронная сеть;
3. Протестировано настольное приложение.