

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Аналитик-разработчик
АО «Тинькофф банк»

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия» профессор
департамента программной
инженерии, канд. техн. наук

_____ Весельев А.Н.
«_____» _____ 2020 г.

_____ В. В. Шилов
«_____» _____ 2020 г.

Модуль для мобильных приложений для определения
эмоционального отклика по изображению пользователя.

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.06-01 81 01-1

Инд. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Исполнитель: студент группы БПИ 199
_____ Д.А. Щербаков
«_____» _____ 2020 г.

Аннотация

Данный документ содержит пояснительную записку к модулю «Facial Expression Recognition Mobile Library» («Модуль для мобильных приложений для определения эмоционального отклика по изображению пользователя»). Модуль служит для определения эмоций человека по изображению его лица, в том числе в режиме реального времени (например, по изображению с камеры). В разделе «Введение» текущего документа указано наименование программы и документы, на основании которых ведется разработка. Раздел «Назначение и область применения» содержит функциональное и эксплуатационное назначение ПО и краткую характеристику области применения. Раздел «Технические характеристики» описывает постановку задачи на разработку, описание алгоритмов и схему функционирования программы, методы организации входных и выходных данных, состав технических и программных средств и обоснование выбора алгоритма, метода организации ввода и вывода и состава технических средств. В разделе «Технико-экономические показатели» отражены предполагаемая потребность и экономические преимущества разработки в сравнении с аналогами. Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.103-77 Обозначения программ и программных документов [2];
3. ГОСТ 19.102-77 Стадии разработки [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Содержание

1	Введение	5
1.1	Наименование программы	5
1.2	Документы, на основании которых ведется разработка	5
2	Назначение и область применения	6
2.1	Назначение программы	6
2.1.1	Функциональное назначение	6
2.1.2	Эксплуатационное назначение	6
2.1.3	Область применения	6
3	Технические характеристики	7
3.1	Постановка задачи на разработку программы и программного модуля . .	7
3.2	Описание алгоритмов и функционирования программного модуля	7
3.2.1	Описание алгоритмов программного модуля	7
3.2.2	Обоснование выбора алгоритмов программного модуля	8
3.2.3	Описание схемы функционирования программного модуля	9
3.3	Описание и обоснование выбора метода организации входных и выходных данных	9
3.3.1	Описание метода организации входных и выходных данных	9
3.3.2	Обоснование выбора метода организации входных и выходных дан- ных	10
3.4	Описание и обоснование выбора состава технических и программных средств	10
3.4.1	Состав технических и программных средств	10
4	Технико-экономические показатели	11
4.1	Предполагаемая потребность	11
4.2	Экономические преимущества разработки и аналоги	11
	Список источников	12
	Приложение А	13
	Приложение В	14
3	Class CameraFeedAnalyzer	14
3.1	Declaration	14
3.2	Поля	14
3.3	Конструктор	14
3.4	Методы	14
3.5	Fields	14
3.6	Constructors	15
3.7	Методы	15
4	Class CameraFeedView	15
4.1	Declaration	15
4.2	Конструктор	15
4.3	Методы	15
4.4	Constructors	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.5	Методы	16
5	Class FacialExpressionViewModel	16
5.1	Declaration	16
5.2	Конструктор	17
5.3	Методы	17
5.4	Constructors	17
5.5	Методы	17

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1 Введение

1.1 Наименование программы

Наименование программы: «Модуль для мобильных приложений для определения эмоционального отклика по изображению пользователя». Краткое наименование – «facialExpressionRecognitionLib».

1.2 Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта «Модуль для мобильных приложений для определения эмоционального отклика по изображению пользователя».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2 Назначение и область применения

2.1 Назначение программы

2.1.1 Функциональное назначение

Модуль позволяет определить эмоции человека на изображении по заданной шкале эмоций.

Прилагаемый к модулю пример эксплуатации позволяет определять эмоциональный отклик на различные записи в социальной сети «Reddit» с помощью данных с камеры пользователя.

2.1.2 Эксплуатационное назначение

Модуль и пример использования должны эксплуатироваться на смартфонах под управлением операционной системы Android.

2.1.3 Область применения

Программа может быть использована в мобильных приложениях для Android в качестве аналитического модуля для улучшения пользовательского опыта или сбора информации о предпочтениях пользователя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3 Технические характеристики

3.1 Постановка задачи на разработку программы и программного модуля

Разработать программный модуль для использования в программах для смартфонов под управлением операционной системы Android с целью анализа эмоций пользователя по его изображению.

Разработать пример эксплуатации модуля в виде программы для смартфона под управлением операционной системы Android, который позволяет определять эмоциональный отклик на различные записи в социальной сети «Reddit» с помощью данных с камеры пользователя.

3.2 Описание алгоритмов и функционирования программного модуля

3.2.1 Описание алгоритмов программного модуля

3.2.1.1 Описание алгоритма работы алгоритма для распознавания эмоций на изображении лица

Для распознавания эмоций по изображению используются сверточные нейронные сети (Convolutional Neural Networks).

Так как создание архитектуры нейронной сети – трудоемкая задача, в основу данного программного модуля легла архитектура, предложенная исследователями Octavio Arriaga, Matias Valdenegro-Toro и Paul Plöger [8]. Она, в свою очередь, опирается на еще более фундаментальные труды в области нейронных сетей и машинного обучения, такие как Xception [9] и Mobilenets [10].

Отличительной особенностью данной архитектуры является ее простота в сравнении со state-of-the-art архитектурами (то есть самыми точными) при довольно высокой точности. Такой результат достигается за счет более эффективной утилизации признаков – например, использование разделенных по глубине сверток или отсутствие полносвязных слоев.

Ниже кратко описаны подходы, используемые в данной нейросети для распознавания эмоций по изображению лица человека.

Разделенные по глубине свертки – метод, при котором для входного сигнала используются две матрицы свертки. Одна поточечно для каждой карты признаков, а другая – в глубину между полученными результатами. Такой подход позволяет сократить количество операций при большем количестве признаков. В приведенной модели данная свертка используется множество раз на тензорах с глубиной 64–128, так что использование разделенной свертки значительно ускоряет время работы сети.

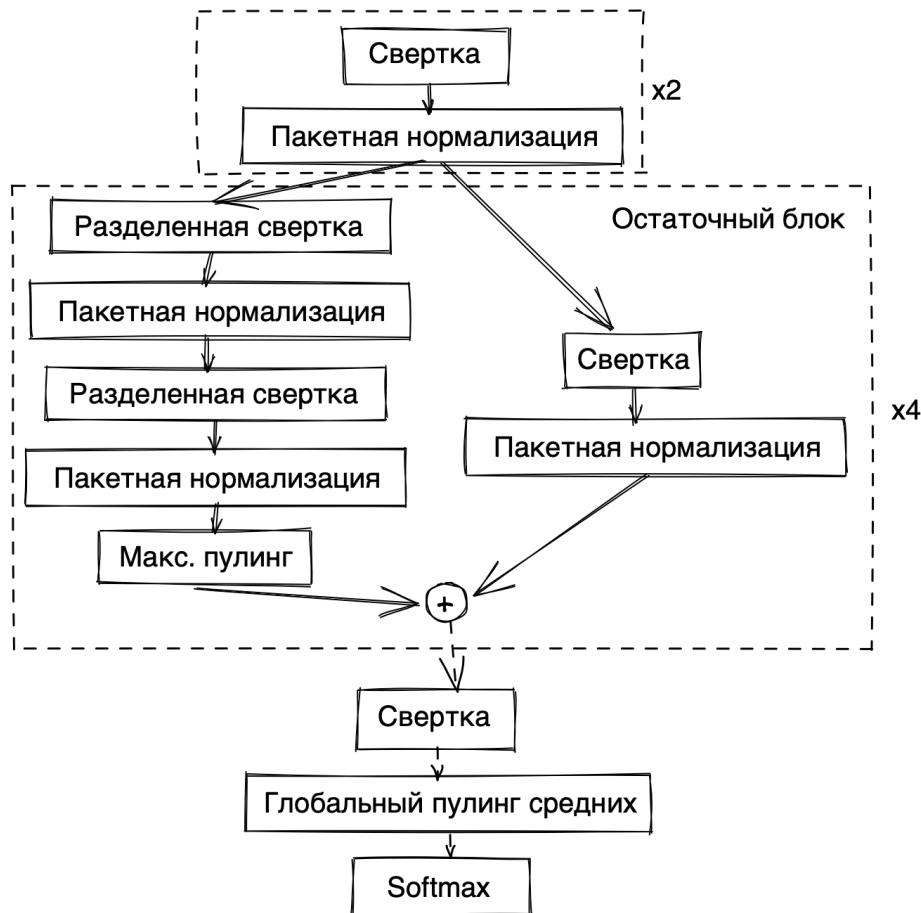
Глобальный пулинг средних (Global average pooling) – операция, обычно используемая для классификации вместо полносвязных слоев. Обычно в последних слоях нейросети используют полносвязные слои, на вход нейронов которых подаются все извлеченные признаки с предыдущего слоя. Такие слои требуют множества вычислений, поэтому для упрощения было предложено использовать метод глобального пулинга – когда последний слой содержит столько карт признаков, сколько мы имеем классов, и затем из каждой карты извлекается среднее и передается в функцию активации.

Остаточные блоки (Residual blocks) – блоки нейросети, используемые для того, чтобы нейросеть при увеличении слоев не теряла точности. В таких блоках резуль-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

тат получается сложением с функцией идентичности $F(x) = H(x) + x$. В современном распознавании изображений с помощью нейронных сетей сложно обойтись без таких блоков [11].

Рисунок 1 — Схема приведенной нейронной сети



На вход нейросети подается изображение в виде тензора формата float размера $1 \times 1 \times 44 \times 44$, соответствующие черно-белой картинке размера 44×44 . На выходе мы получаем тензор размера $1 \times 1 \times 1 \times 7$ и извлекаем из него комбинации из 7 различных эмоций.

Для обучения сети используется метод оптимизации «стохастический градиентный спуск» и функция для минимизации «перекрестная энтропия». Это довольно стандартные методы для машинного обучения классификаторов с заданным количеством классов.

3.2.2 Обоснование выбора алгоритмов программного модуля

3.2.2.1 Обоснование выбора алгоритма для распознавания эмоций на изображении лица

Сверточные нейронные сети являются очень эффективным инструментом для распознавания различных характеристик на изображениях.

Выбор в пользу конкретной архитектуры был сделан из-за ее высокого соотношения точности на программную сложность. Малое количество слоев и параметров и отсутствие полносвязных слоев позволяет быстрее реализовать и обучить нейросеть, а также ускоряет ее работу и уменьшает требуемый для работы размер оперативной памяти устройства.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Скорость работы и объем используемой памяти очень критичны для мобильных устройств, имеющих слабые процессоры и работающих от аккумуляторной батареи. На устройстве Huawei Honor 10 с процессором Kirin 970 и при использовании аппаратного ускорения нейронных сетей (технология NNAPI) нейросеть обрабатывает изображение размером 44x44 пикселя за менее чем 50 мс, что является удовлетворительным результатом для работы в режиме реального времени.

3.2.3 Описание схемы функционирования программного модуля

Обученная модель компилируется в JIT-оптимизированный формат вычислительных программ TorchScript [12], который можно запустить с помощью библиотеки pytorch_android. На устройстве под управлением Android такие скрипты выполняются на CPU и по возможности используют модуль нейронных сетей с помощью NNAPI [13].

Для выделения лица на фото используется библиотека MLKit, которая также работает с помощью нейросетей. Данная библиотека предоставляет высокоуровневое API для анализа изображений в асинхронном режиме и различными методами распознавания объектов, и в рамках данного проекта углубляться в архитектуру этой библиотеки не выглядит целесообразным.

После выделения прямоугольника, в котором находится лицо, необходимо извлечь эту часть изображения и преобразовать ее в тензор, который будет использоваться моделью. Изображения с камеры на смартфонах под управлением Android поступают в формате YUV_420_888, что означает, что каждое изображение состоит из трех буферов, которые с помощью специальных алгоритмов объединяются для вывода на экран. В случае приведенной модели используется черно-белое изображение, поэтому достаточно извлечь только информацию из первого буфера, или Y-буфера, в котором содержится яркость каждого отдельного пикселя, и перевести это число в формат float, нормализовав по модулю 255. Выбираются только те пиксели, что содержат лицо, и ужимаются в небольшой квадрат размером 44x44 пикселя с помощью операций сжатия и растяжения. Далее с помощью библиотеки Pytorch тензор передается в модель, и на выходе получается вектор длины 7, в каждой ячейке которого содержится вес определенной эмоции. Впоследствии этот вектор можно интерпретировать по-разному, например, выбрать самое большое число в качестве главной эмоции, или выбрать только те значения, что выше какого-либо порога. В приведенном примере программы выбирается наибольшее число и конвертируется в строку – например, первому значению соответствует эмоция "Злость".

3.3 Описание и обоснование выбора метода организации входных и выходных данных

3.3.1 Описание метода организации входных и выходных данных

В модуль на вход может подаваться изображение в формате YUV_420_888. Также в модуле имеются средства для инициализации камеры и непосредственного анализа информации с нее.

В качестве выходных данных модуль передает вектор длины 7 с числами от 0 до 1, каждое из которых соответствует вероятности распознанной эмоции из набора стандартных эмоций.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3.2 Обоснование выбора метода организации входных и выходных данных

Формат YUV_420_888 является основным стандартом изображений с камеры для смартфонов на ОС Android.

Полученный с помощью Softmax вектор с весами различных классов – стандартная практика при решении задач классификации.

3.4 Описание и обоснование выбора состава технических и программных средств

3.4.1 Состав технических и программных средств

Для работы программного модуля необходим следующий набор программных средств:

1. операционная система Android версии 8.0 и выше

Для работы программного модуля необходим следующий состав технических средств:

1. Не менее 512МБ ОЗУ;
2. Не менее 150МБ свободного места на внутреннем накопителе;

Минимальное количество памяти ОЗУ, необходимое для работы системы Android 8.0 и выше составляет 512МБ [14].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4 Технико-экономические показатели

4.1 Предполагаемая потребность

Данный программный модуль позволит разработчикам приложений для Android быстро и легко решить задачу распознавания эмоций пользователя.

4.2 Экономические преимущества разработки и аналоги

Преимущества данного модуля заключаются в быстродействии анализатора, а также в легкой интеграции с программами для ОС Android. Других общедоступных библиотек, включающих в себя все функции данного модуля, такие как: предварительная обработка изображения, подключение к камере и запуск модели, найдено не было. Экономическая выгода использования данной разработки заключается в экономии времени разработчиков приложений.

	Подготовка изображения	Подключение камеры	Анализ изображения	Распознавание эмоций	Быстродействие
PyTorch Android	-	-	+	-	Зависит от модели
TensorFlow Lite	-	-	+	-	Зависит от модели
Google MLKit	-	+	+	-	+
facialExpression RecognitionLib	+	+	+	+	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Список источников

- [1] “ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [2] “ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [3] “ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [4] “ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [5] “ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [6] “ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [7] “ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001..”
- [8] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, “Real-time convolutional neural networks for emotion and gender classification,” *CoRR*, vol. abs/1710.07557, 2017.
- [9] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2017.
- [10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [12] “Torchscript [Электронный ресурс].” // URL: <https://pytorch.org/docs/stable/jit.html>. (Дата обращения: 17.05.2021, режим доступа: свободный).
- [13] “Neural networks api [Электронный ресурс].” // URL: <https://developer.android.com/ndk/guides/neuralnetworks>. (Дата обращения: 17.05.2021, режим доступа: свободный).
- [14] “Android 8.0 compatibility definition [Электронный ресурс].” // URL: <https://source.android.com/compatibility/8.0/android-8.0-cdd.pdf>. (Дата обращения: 17.05.21, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ А

Используемые понятия и определения

- Тензор
- Сверточная нейронная сеть
- Функция активации
- Формат изображений YUV_420_888

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Описание и функциональное назначение классов и структур

Пакет ru.hse.dascherbakov_1.facialExpressionRecognition

Классы

CameraFeedAnalyzer	14
Класс анализатора изображения	
CameraFeedView	15
Фрагмент для предпросмотра изображения с камеры и анализа эмоций по изображению лица в реальном времени	
FacialExpressionViewModel	16

Class CameraFeedAnalyzer

Класс анализатора изображения

3.1 Declaration

```
public class CameraFeedAnalyzer  
    extends java.lang.Object
```

3.2 Поля

классы

3.3 Конструктор

CameraFeedAnalyzer(CameraFeedView)

3.4 Методы

analyze(ImageProxy) Основной метод для анализа
buildAnalysis(CameraFeedView) Получить анализатор для подключения к камере

3.5 Fields

– public static final java.lang.String[] классы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.6 Constructors

– CameraFeedAnalyzer

```
public CameraFeedAnalyzer(CameraFeedView cameraFeedView)
```

3.7 Методы

– analyze

```
public void analyze(ImageProxy imageProxy)
```

– Описание

Основной метод для анализа

– buildAnalysis

```
public static ImageAnalysis buildAnalysis(CameraFeedView cameraFeedView)
```

– Описание

Получить анализатор для подключения к камере

Class CameraFeedView

Фрагмент для предпросмотра изображения с камеры и анализа эмоций по изображению лица в реальном времени

4.1 Declaration

```
public class CameraFeedView  
    extends Fragment
```

4.2 Конструктор

CameraFeedView()

4.3 Методы

getViewModel() Получение класса ViewModel

onActivityCreated(Bundle) Завершение создания фрагмента

onCreateView(LayoutInflater, ViewGroup, Bundle) Инициализация визуальных виджетов фрагмента

onViewCreated(View, Bundle) Метод, выполняемый после инициализации визуальных виджетов фрагмента

setOnClickListener(View.OnClickListener)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.4 Constructors

– CameraFeedView

```
public CameraFeedView()
```

4.5 Методы

– getViewModel

```
public FacialExpressionViewModel getViewModel()
```

– Описание

Получение класса ViewModel

– onActivityCreated

```
public void onActivityCreated(Bundle savedInstanceState)
```

– Описание

Завершение создания фрагмента

– onCreateView

```
public View onCreateView(LayoutInflater inflater, ViewGroup viewGroup, Bundle savedInstanceState)
```

– Описание

Инициализация визуальных виджетов фрагмента

– onViewCreated

```
public void onViewCreated(View view, Bundle savedInstanceState)
```

– Описание

Метод, выполняемый после инициализации визуальных виджетов фрагмента

– setOnClickListener

```
public void setOnClickListener(View.OnClickListener listener)
```

Class FacialExpressionViewModel

5.1 Declaration

```
public class FacialExpressionViewModel  
    extends ViewModel
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5.2 Конструктор

FacialExpressionViewModel()

5.3 Методы

getFaceRect()
getImageSize()
getLabel()
getOutput()
setFaceRect(RectF)
setImageSize(Size)
setLabel(String)
setOutput(float[])

5.4 Constructors

- **FacialExpressionViewModel**

```
public FacialExpressionViewModel()
```

5.5 Методы

- **getFaceRect**

```
public <any> getFaceRect()
```

- **getImageSize**

```
public <any> getImageSize()
```

- **getLabel**

```
public <any> getLabel()
```

- **getOutput**

```
public <any> getOutput()
```

- **setFaceRect**

```
public void setFaceRect(RectF faceRect)
```

- **setImageSize**

```
public void setImageSize(Size imageSize)
```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
ИНВ. № ПОДЛ.	Подп. и дата	Взам. инв. №	ИНВ. № дубл.	Подп. и дата

– **setLabel**

```
public void setLabel(java.lang.String label)
```

– **setOutput**

```
public void setOutput(float [] output)
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.06-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата