

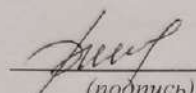
Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

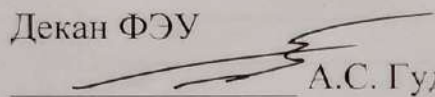
Работа выполнена в СКБ «Электроника и робототехника»

СОГЛАСОВАНО

Начальник отдела ОНИПКРС

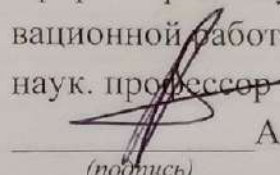
  
(подпись) Е.М. Димитриади  
« 23 » 01 2024 г.

Декан ФЭУ

  
(подпись) А.С. Гудим  
« 23 » 01 2024 г.

УТВЕРЖДАЮ

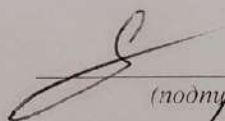
Проректор по научной и инно-  
вационной работе, д-р техн.  
наук. профессор

  
(подпись) А.В. Космынин  
« 23 » 01 2024 г.

«ЕмоСruit: Эмоциональный интеллект в рекрутинге»

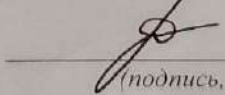
Комплект конструкторской документации

Руководитель СКБ

  
(подпись, дата)

В.В. Солецкий

Руководитель проекта

  
(подпись, дата)

Р.В. Шибeko

Комсомольск-на-Амуре 2023



Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

ЗАДАНИЕ  
на разработку

Название проекта: EmoCruit: Эмоциональный интеллект в рекрутинге

Назначение: Распознавание эмоционального интеллекта в рекрутинге, позволяет при отборе HR-собеседовании рекрутеру. Этот инновационный подход позволяет лучше понять невербальные сигналы, сократить временные издержки и повысить точность отбора, что критически важно для оптимизации процесса подбора персонала и успешного развития компаний.

Область использования: B2C

Функциональное описание проекта: Проект по распознаванию мимики на онлайн-собеседованиях - это способ использования компьютеров, чтобы помочь нам понимать, как люди чувствуют себя во время интервью или разговора через видеосвязь. Когда мы говорим с кем-то, наше лицо выражает разные эмоции: улыбка, грусть, удивление и так далее. Эти изменения на лице называются мимикой.

Идея этого проекта заключается в том, чтобы обучить компьютеры "читать" мимику так же, как мы это делаем. Для этого нужно научить их понимать, какие движения лица соответствуют разным эмоциям. Когда человек улыбается, его глаза могут зажечься, а уголки рта подниматься. Когда он грустит, лицо может стать меньше выразительным.

Техническое описание устройства: Инновационный программный продукт, который обеспечит автоматизированное распознавание мимики и выражений лица на основе видеоданных, предоставляя ценные и объективные сведения для более эффективного и точного процесса подбора персонала.

Основным компонентом продукта будет алгоритм обработки изображений и анализа видео, разработанный на базе машинного обучения и искусственного интеллекта. Этот алгоритм будет обучен распознавать ключевые движения и морщины лица, связанные с разными эмоциями, такими как радость, грусть, удивление, страх и т.д.

Программный продукт будет иметь простой и интуитивно понятный интерфейс, который позволит пользователям легко загружать видеоматериалы с интервью кандидатов. После обработки видео продукт будет генерировать детализированный анализ эмоциональных реакций кандидатов в ходе собеседования. Анализ результатов будет представлен в виде дашборда или отчета.

Требования: Платформа должна обеспечивать конфиденциальность личностей и корректно работать.

План работ:

Наименование работ	Срок
Сбор данных	10.2023
Обучение алгоритма	11.2023
Разработка программного продукта	12.2023
Тестирование и улучшение	01.2024


Комментарии:

---

Перечень графического материала:

1. Блок-схема работы изделия;
  2. Функциональная схема;
  3. Внешний вид изделия, результат его работы;
- 
- 

Руководитель проекта

---

(подпись, дата)

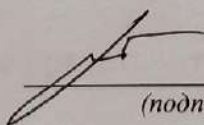
Р.В. Шибeko

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

**ПАСПОРТ**

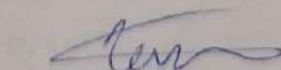
**«ЕмоСruit: Эмоциональный интеллект в рекрутинге»**

Руководитель проекта

  
\_\_\_\_\_  
(подпись, дата)

Р.В. Шибeko

Исполнители проекта

  
\_\_\_\_\_  
(подпись, дата)

Е.С. Ильченко

## Содержание

1	Общие положения .....	7
1.1	Наименование изделия .....	7
1.2	Наименования документов, на основании которых ведется проектирование изделия.....	7
1.3	Перечень организаций, участвующих в разработке изделия .....	7
1.4	Сведения об использованных при проектировании нормативно-технических документах .....	8
2	Назначение и принцип действия .....	9
2.1	Назначение изделия .....	9
2.2	Области использования изделия .....	9
2.3	Принцип действия изделия .....	9
3	Состав изделия и комплектность.....	10
4	Технические характеристики.....	11
4.1	Основные технические характеристики блока	
4.2	Основные технические характеристики	
5	Устройство и описание работы изделия .....	12
5.1	Устройство изделия .....	12
5.2	Описание работы изделия .....	17
6	Условия эксплуатации .....	19
6.1	Правила и особенности размещения изделия .....	19
6.2	Меры безопасности.....	19
6.3	Правила хранения и транспортирования.....	20
	ПРИЛОЖЕНИЕ А .....	7
	ПРИЛОЖЕНИЕ Б.....	7

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		6

## **1 Общие положения**

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «EmoCruit: Эмоциональный интеллект в рекрутинге» (далее «изделие»).

Паспорт входит в комплект поставки изделия. Прежде, чем пользоваться изделием, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию изделия, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

### **1.1 Наименование изделия**

Полное наименование изделия – «EmoCruit: Эмоциональный интеллект в рекрутинге» (EmoCruit).

### **1.2 Наименования документов, на основании которых ведется проектирование изделия**

Проектирование «EmoCruit: Эмоциональный интеллект в рекрутинге» осуществляется на основании требований и положений следующих документов:

- задание на разработку.

### **1.3 Перечень организаций, участвующих в разработке изделия**

Заказчиком проекта «EmoCruit: Эмоциональный интеллект в рекрутинге» является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 17.

Исполнителями проекта «EmoCruit» являются Конструкторы студенческого конструкторского бюро «Электроника и робототехника» (далее СКБ), студент группы ОБМБ-1 Ильченко Елизавета Сергеевна.

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		7

#### **1.4 Сведения об использованных при проектировании нормативно-технических документах**

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации. Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации. Правила выполнения эксплуатационных документов.

ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации. Электронные документы. Общие положения.

ГОСТ 2.052-2006. Единая система конструкторской документации. Электронная модель изделия. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации. Эксплуатационные документы.

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		8



## **2 Назначение и принцип действия**

### **2.1 Назначение изделия**

EmoCruit: Эмоциональный интеллект в рекрутинге – предназначен для улучшения процесса подбора персонала путем автоматизированного распознавания эмоционального состояния кандидатов во время онлайн собеседований. EmoCruit обеспечивает более эффективное использование времени рекрутеров, фокусируясь на невербальных сигналах, для лучшего понимания эмоциональных реакций кандидатов. Цель продукта - сократить временные издержки, предоставляя рекрутерам ценные и объективные данные для более эффективного процесса подбора персонала.

Программный продукт предоставляет простой интерфейс для загрузки видеоматериалов и генерации детализированных анализов эмоциональных реакций кандидатов.

### **2.2 Области использования изделия**

Изделие может применяться в компании, производстве и для личного пользования.

### **2.3 Принцип действия изделия**

Принцип действия "EmoCruit" основан на использовании программно-аппаратного комплекса на базе Raspberry Pi 4. Камера и микрофон, подключенные к устройству, записывают видео и аудио во время онлайн собеседований. Алгоритм обработки изображений, обученный на основе машинного обучения, анализирует видеопоток для распознавания мимики и выражений лица, выявляя эмоциональные реакции кандидатов. В то же время, алгоритм распознавания голоса анализирует аудиозапись, выделяя интонации и акустические параметры, свидетельствующие о эмоциональном состоянии. Результаты анализа интегрируются, создавая комплексную картину эмоционального интеллекта кандидата, предоставляя рекрутерам ценные и объективные данные для более эффективного отбора персонала.

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		9

### 3 Состав изделия и комплектность

В комплект поставки входит:

- Программно-аппаратный комплекс;
- Зарядка;
- Дисплей;
- Руководство для пользования;
- Паспорт.

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		10

#### 4 Технические характеристики

Основные технические характеристики приведены в таблице 1.

Таблица 1 – Технические характеристики изделия

Наименование параметра	Значение
Графический процессор	GPU
CUDA-ядра	256
Разрешение камеры, пикс.	1920 x 1280
Процессор	Cortex-A72 (ARMv8-A) 64-бит
Операционная система	Linux-дистрибутив, ubuntu
Скорость распознавания объекта на кадре, с	0,1
Порты	USB, Ethernet
Интерфейсы	UART, GPIO
Питание, В	5
Масса нетто, кг	1

## 5 Устройство и описание работы изделия

### 5.1 Устройство изделия

Платформа работает с помощью мультимодального искусственного интеллекта. Используется модуль временного внимания (TAM) для поиска ключевых кадров, связанных с обнаружением стресса в выражениях лица. TAM присваивает разные обучаемые веса различным фреймам выражений лица, чтобы выделить отличительную временную репрезентацию выражений лица, свидетельствующих о стрессе. Фреймворк глубокого обучения, который сочетает голос и выражения лица для обнаружения острого стресса. Метод слияния основан на собственном векторе матрицы, который обеспечивает точность обнаружения 85,1%. Датасет взят из эксперимента MIST, который включал в себя видеокамеру Sony FDR-AX700. Камера фиксировала выражение лиц и голоса участников во время эксперимента MIST и отправлял их на собирающий компьютер, где разрешение видео составляет 1920 \* 1080 и 30 кадров в секунду. На рисунке 5.1 представлена схема сбора датасета.

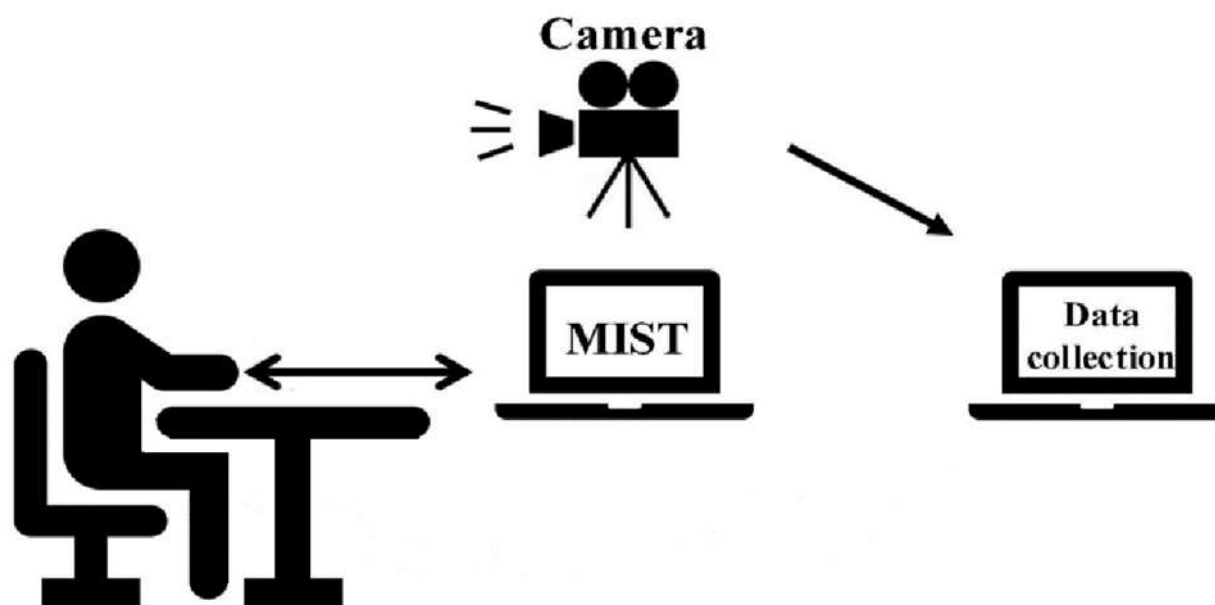


Рисунок 5.1 – Схема сбора датасета

Предварительная обработка выражения лица. Удалена была фоновая информация, чтобы изолировать область лица, которая может не быть нарушена нерелевантной информацией об окружающем шуме и беспорядке в

различных сценариях реальности. Видео  $V_i$  было оформлено в последовательности кадров изображения  $V_i=(frame_{i1}, frame_{i1}, \dots, frame_{in})$ , а область лица была обнаружена МТСNN на каждом кадре изображения и затем выровнено.  $Face_i=(face_{i1}, face_{i1}, \dots, face_{in})$  обозначает последовательность изображений лиц, обнаруженных из последовательности  $V_i$ .

Предварительная обработка голоса, представляла из себя спектограмму Mel, рассчитанная в частотно-временном анализе и содержащая информацию о времени и частоте голоса. Он преобразует линейную шкалу частот в логарифмическую шкалу и представляет распределение энергии сигнала на частоте мелкомасштабной шкалы, что аналогично человеческому слуху. Спектрограмма Mel может интуитивно показывать спектральные изменения голоса с течением времени.

После разделения голосовых данных на каждую выборку выделялись данные с предварительным акцентом. Затем окно Хэмминга длиной 30 мс использовалась для формирования данных с перекрытием в 15 мс.

После этого мы вычислили плотность энергии с помощью кратковременного преобразования Фурье (STFT) и преобразовали частоту в диапазон мелкомасштабных значений для получения спектрограммы Mel.

Модель глубокого обучения разработана путем объединения ResNet50 и I3D с модулем временного внимания. ResNet50 извлекает глобальные и локальные характеристики спектрограммы Mel посредством сопоставления идентификационных данных. I3D с модулем временного внимания изучает пространственно-временные изменения в выражениях лица, а модуль временного внимания позволяет I3D извлекать важные временные характеристики.

Информация о напряженном состоянии в их полностью связанных слоях была объединена в глобальную матрицу, что привело к мультимодальному представлению информации о напряжении. Мультимодальная информация о

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		13

стрессе использовалась для определения стресса на основе собственных векторов матрицы

Преобразование голосовых сигналов в трехмерные матрицы представляет сигналы более высокого порядка и нелинейные характеристики. В этой работе использовался Resnet в 2D-CNN для извлечения характеристик из спектрограммы Mel. ResNet позволяет избежать проблемы исчезновения градиента и резкого увеличения в традиционном 2D-CNN за счет коротких подключений. Стратегия 2D-свертки входных данных Mel-спектрограмм в этой работе представлена на рисунке 5.2

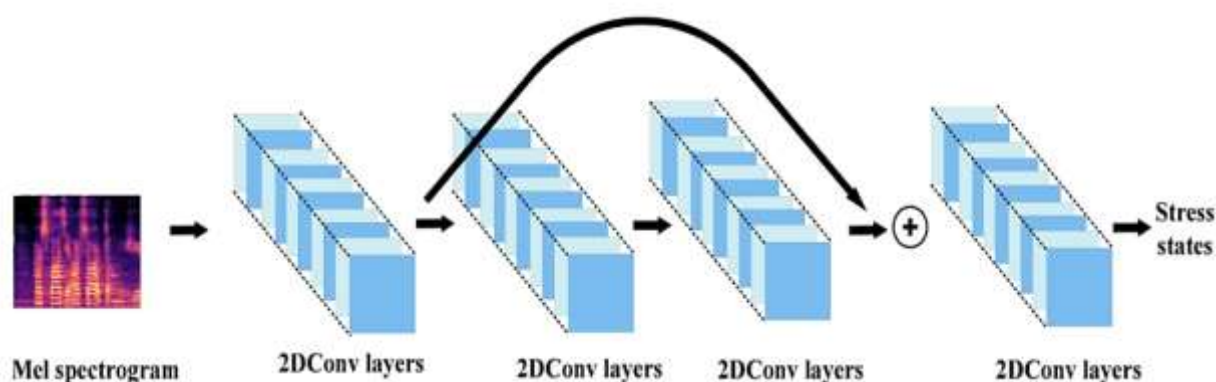


Рисунок 5.2 – Стратегия 2D-свертки входных данных Mel-спектрограмм

Стратегия 2D-свертки для ввода спектрограммы Mel. 2D-CNN имеет слой свертки, состоящий из ядер 2D-свертки. Сверточный уровень может извлекать объекты в нескольких измерениях, чтобы получить представление внутренней структуры спектрограммы Mel, сканируя их с помощью ядер двумерной свертки и уменьшая количество параметров за счет локального подключения и совместного использования параметров. Двумерная свертка может быть выражена следующим образом:

$$v_{ij}^{xy} = f(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} W_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)})$$

где  $v_{ij}^{xy}$  - результат свертки  $i$  в точке  $j$  на карте объектов  $(x,y)$  слоя;  $f$  - функция активации, выпрямленная линейной единицей (ReLU);  $b_{ij}$  - отклонение карты

объектов;  $m$  - индекс карты объектов в слое  $i-1$ ;  $Q_i, P_i$  - высота и ширина ядра свертки;  $W_{ijm}^{pq}$  - представляет значение в позиции карты объектов.

На рисунке 5.3 представлена общая архитектура ResNet50.

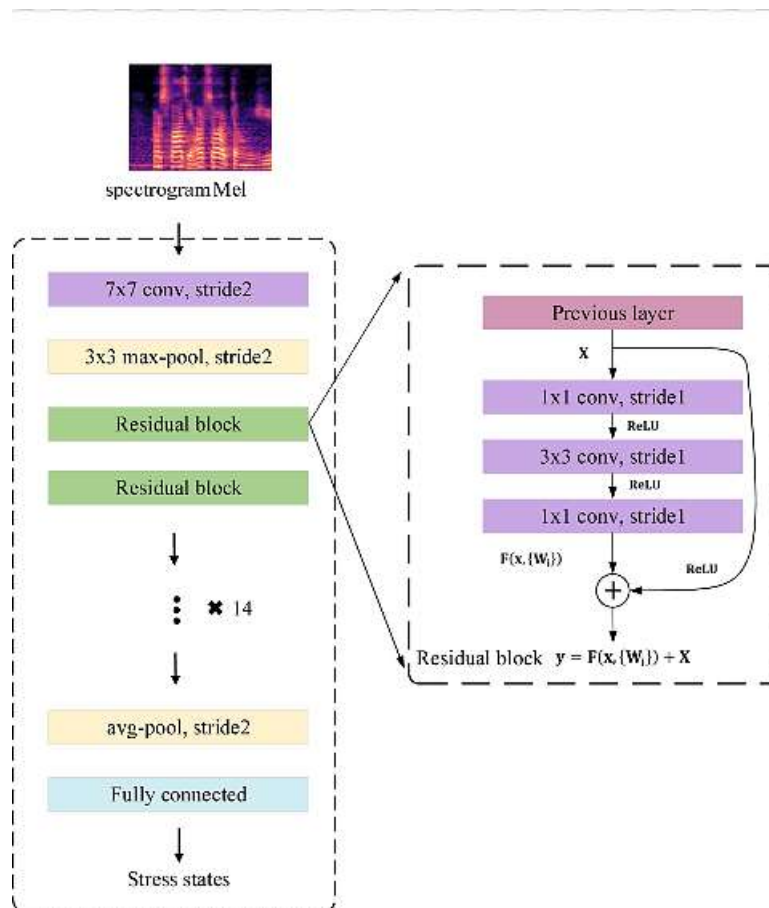


Рисунок 5.3 - Общая архитектура ResNet50.

$$y = F(x, \{W_i\}) + X$$

где  $X$  - входные данные весового слоя; ReLU - функция активации;  $F(x, \{W_i\})$  - выходные данные после трех слоев свертки. Отображение идентичности добавляет  $F(X)$  и  $X$  в качестве входных данных  $y$  в следующий остаточный блок.

Определение стресса по выражению лица требует сравнения временных изменений в нескольких областях лица, включая глаза, рот и щеки. Использовался расширенный 3D-CNN (I3D) с модулем временного внимания для изучения общих изменений выражений лица во время стресса. I3D извлекает

признаки стресса из выражений лица, а модуль временного внимания сообщает I3D, какие фреймы важны, представлено на рисунке 5.4.

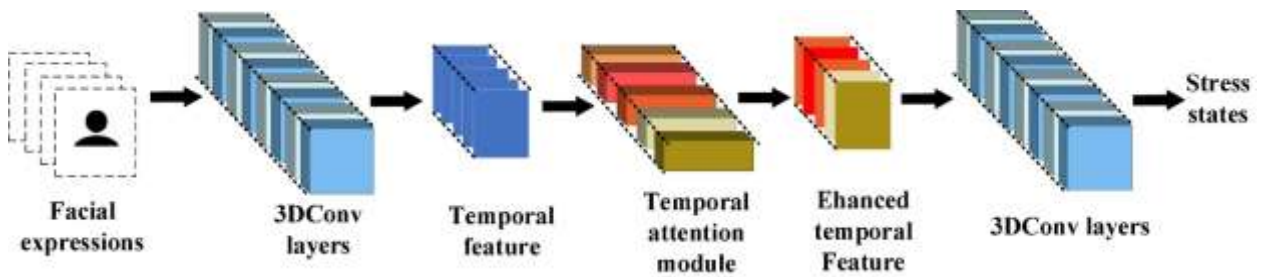


Рисунок 5.4 - Стратегия 3D-свертки для ввода выражений лица

3D-CNN имеет ядро 3D-свертки, которое может анализировать последовательные кадры выражений лица и фиксировать пространственно-временные особенности выражений лица. 3D-свертку можно выразить как:

$$v_{km}^{xyz} = f(b_{km} + \sum_{p=0}^{P_k-1} \sum_{q=0}^{Q_k-1} \sum_{r=0}^{R_k-1} W_{kmn}^{pqr} u_{(k-1)n}^{(x+p)(y+q)(z+r)})$$

где  $v_{km}^{xyz}$  - является позицией k на карте объектов m слоя (x,y,z); f – функция потерь; u - входные данные от уровня от k-1 до k;  $P_k, Q_k, R_k$  - являются шириной высотой и глубиной размера ядра свертки;  $b_{km}$  - это отклонение карты объектов  $W_{kmn}^{pqr}$ .

На рисунке 5.5 представлен метод слияния, основанный на собственном векторе матрицы.



Рисунок 5.5 – Метод слияния

На рисунке 5.6 представлена общая архитектура I3D с ТАМ.



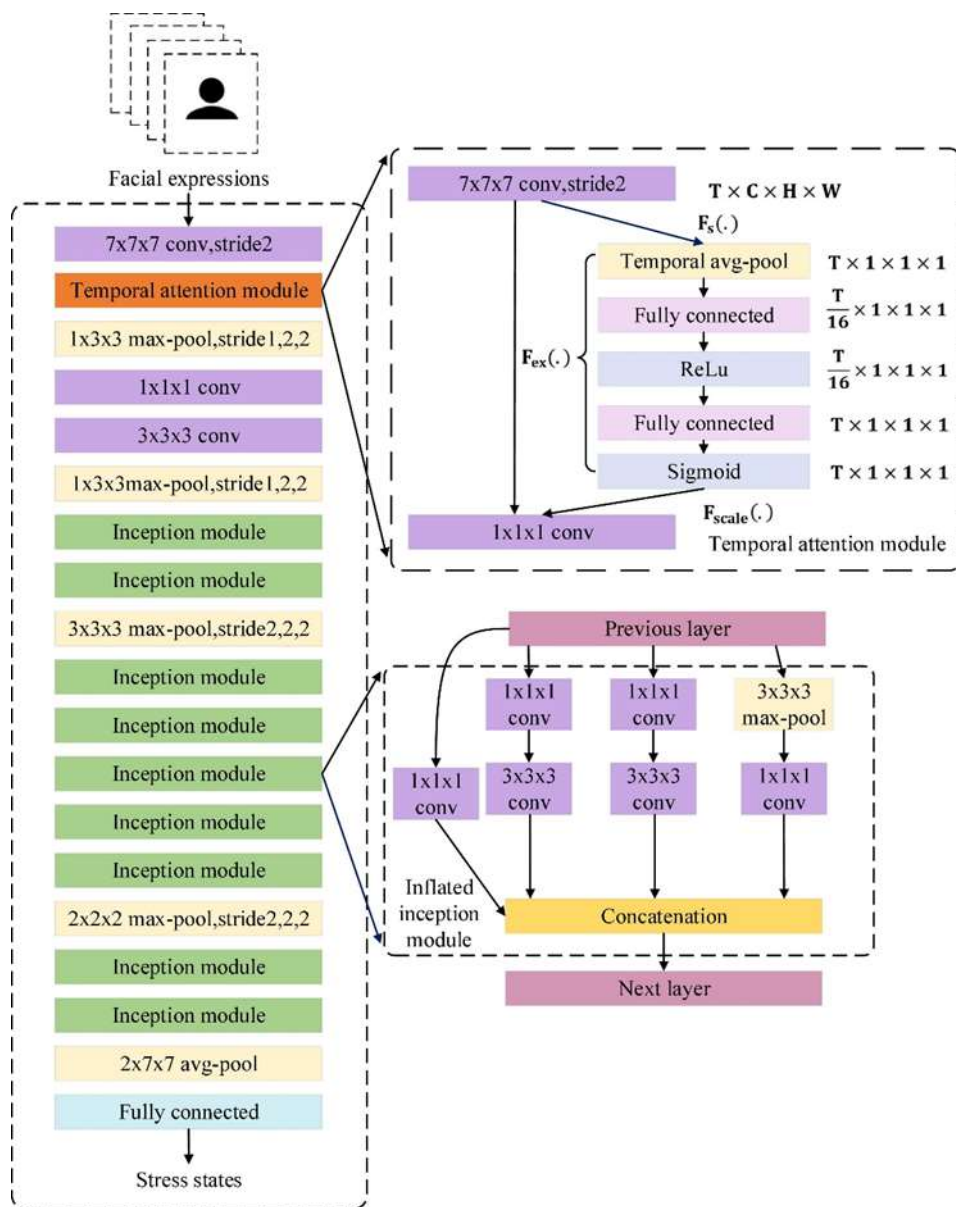


Рисунок 5.6 - Общая архитектура I3D с TAM

Функциональная схема представлена в Приложении А.

## 5.2 Описание работы изделия

Фреймворк глубокого обучения на Raspberry pi 4 объединил голос и мимику для определения стресса. Информация о стрессе в мультимодальности может быть получена с полностью подключенного уровня ResNet50 и I3D с модулем временного внимания в фреймворке. Фреймворк интегрировал их в

глобальную матрицу для представления и использовал метод слияния, основанный на собственных векторах матрицы, для обнаружения стресса.

Мультимодальный метод обеспечил наилучшую производительность при обнаружении стресса, предполагая, что фреймворк глубокого обучения, использующий мультимодальные данные для обнаружения стресса, может обеспечить лучшую производительность при обнаружении стресса, чем метод, основанный на одной модальности.

Как показано в таблице 1, обнаружение стресса с использованием мультимодальности повысило производительность по сравнению с использованием только одномодальных данных. Точность мультимодального результата повышена по сравнению с максимальной точностью одномодального результата, составляющей 83,9–85,1%.

Таблица 1 - Точность определения стресса, выверенность, отзыв и оценка F1 с использованием одномодальных и мультимодальных данных.

Modal	Accuracy	Precision	Recall	F1 - Score
Voice	0.830	0.825	0.829	0.827
Facial exp.	0.792	0.795	0.803	0.799
Fusion	0.851	0.857	0.866	0.861

Использование программно-аппаратного комплекса, подключается питание к raspberry, и запускает сервер, который отслеживает состояние сотрудников.

Программный код представлен в Приложении Б.

## 6 Условия эксплуатации

Изделие выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется изделие не должно возникать условий для конденсации влаги (выпадения росы). Изделие является электронным прибором, требующим бережного обращения.

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с изделием;
- предохранять изделие от ударов и повреждений;
- не допускать самостоятельную разборку изделия.

### 6.1 Правила и особенности размещения изделия

Изделие должно быть расположено на расстоянии не менее 1 м от нагревательных приборов.

**ВНИМАНИЕ!** При эксплуатации изделия запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов изделия.

### 6.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- не оставлять изделие включенным без наблюдения;
- после транспортировки в холодное время года изделие необходимо выдержать при комнатной температуре не менее двух часов;

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		19

- внутренние осмотры и ремонт изделия должны производиться только квалифицированными специалистами;
- не устанавливайте изделие на неустойчивой подставке, стойке или ненадежном кронштейне.

### **6.3 Правила хранения и транспортирования**

Транспортирование изделия в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования изделия по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

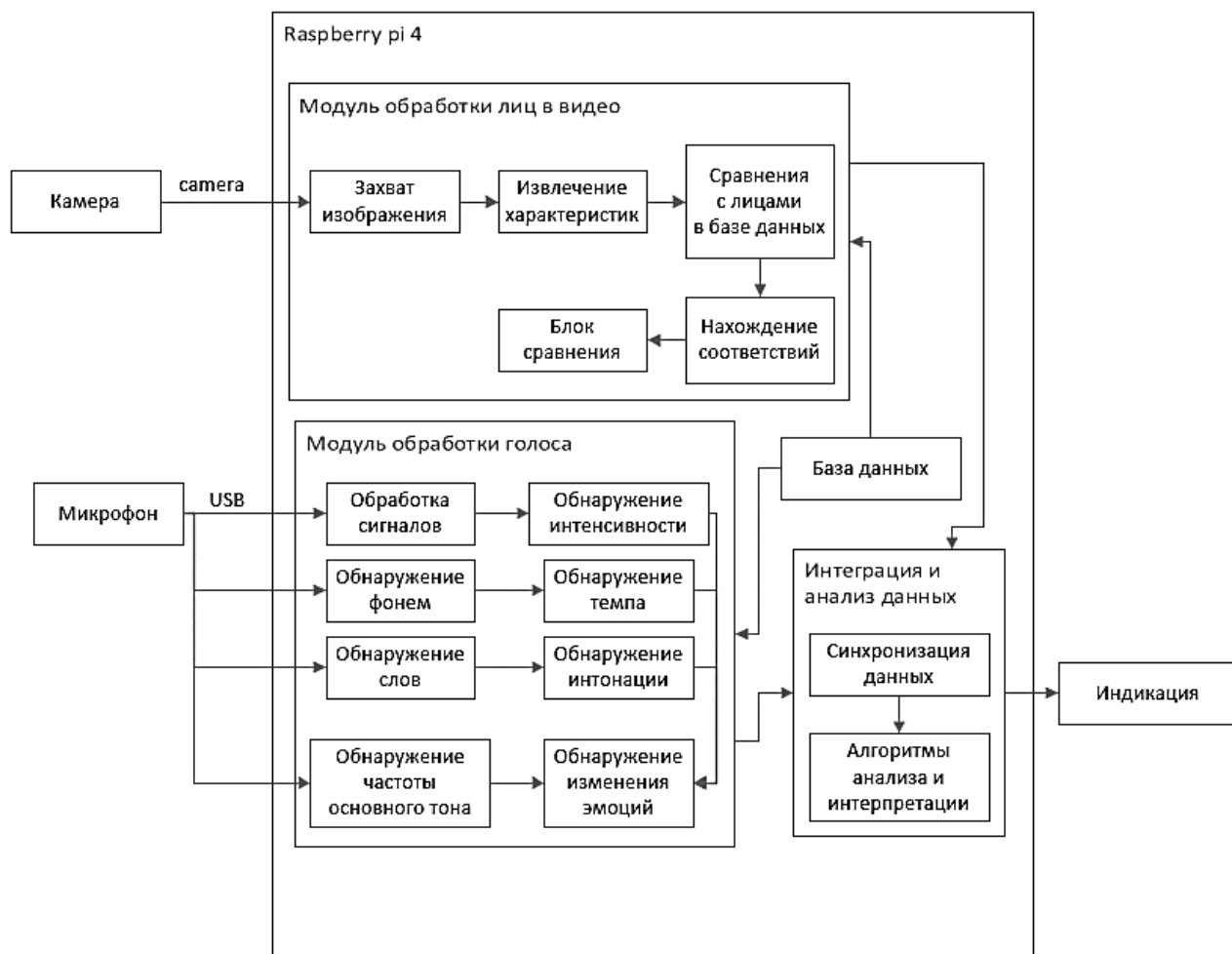
После транспортирования изделие должно быть выдержано не менее 2 часов в транспортной таре при температуре  $20 \pm 5$  °С и относительной влажности воздуха не более 80 %.

Распакованное изделие должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от +5 до +40 °С и относительной влажности воздуха не более 60 %. При температуре ниже 25 °С допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		20

# ПРИЛОЖЕНИЕ А

(обязательное)



## ПРИЛОЖЕНИЕ Б

(обязательное)

```
#Модуль распознавание стресса по речи
from emotion_recognition import EmotionRecognizer
import pyaudio
import os
import wave
from sys import byteorder
from array import array
from struct import pack
from sklearn.ensemble import GradientBoostingClassifier, BaggingClassifier
from utils import get_best_estimators
THRESHOLD = 500
CHUNK_SIZE = 1024
FORMAT = pyaudio.paInt16
RATE = 16000
SILENCE = 30
def is_silent(snd_data):
    "Returns 'True' if below the 'silent' threshold"
    return max(snd_data) < THRESHOLD
def normalize(snd_data):
    "Average the volume out"
    MAXIMUM = 16384
    times = float(MAXIMUM)/max(abs(i) for i in snd_data)
    r = array('h')
    for i in snd_data:
        r.append(int(i*times))
    return r
```

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		22

```

def trim(snd_data):
    "Trim the blank spots at the start and end"
    def _trim(snd_data):
        snd_started = False
        r = array('h')
        for i in snd_data:
            if not snd_started and abs(i)>THRESHOLD:
                snd_started = True
                r.append(i)

            elif snd_started:
                r.append(i)
        return r
    # Trim to the left
    snd_data = _trim(snd_data)
    # Trim to the right
    snd_data.reverse()
    snd_data = _trim(snd_data)
    snd_data.reverse()
    return snd_data

def add_silence(snd_data, seconds):
    "Add silence to the start and end of 'snd_data' of length 'seconds' (float)"
    r = array('h', [0 for i in range(int(seconds*RATE))])
    r.extend(snd_data)
    r.extend([0 for i in range(int(seconds*RATE))])
    return r

def record():
    """"
    Record a word or words from the microphone and

```

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		23

return the data as an array of signed shorts.  
Normalizes the audio, trims silence from the start and end, and pads with 0.5 seconds of blank sound to make sure VLC et al can play it without getting chopped off.

"""

```
p = pyaudio.PyAudio()
stream = p.open(format=FORMAT, channels=1, rate=RATE,
                input=True, output=True,
                frames_per_buffer=CHUNK_SIZE)
num_silent = 0
snd_started = False
r = array('h')
while 1:
    # little endian, signed short
    snd_data = array('h', stream.read(CHUNK_SIZE))
    if byteorder == 'big':
        snd_data.byteswap()
    r.extend(snd_data)
    silent = is_silent(snd_data)
    if silent and snd_started:
        num_silent += 1
    elif not silent and not snd_started:
        snd_started = True
    if snd_started and num_silent > SILENCE:
        break
sample_width = p.get_sample_size(FORMAT)
stream.stop_stream()
stream.close()
```

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		24



```

p.terminate()
r = normalize(r)
r = trim(r)
r = add_silence(r, 0.5)
return sample_width, r
def record_to_file(path):
    "Records from the microphone and outputs the resulting data to 'path'"
    sample_width, data = record()
    data = pack('<' + ('h'*len(data)), *data)
    wf = wave.open(path, 'wb')
    wf.setnchannels(1)
    wf.setsampwidth(sample_width)
    wf.setframerate(RATE)
    wf.writeframes(data)
    wf.close()
def get_estimators_name(estimators):
    result = [ "{}".format(estimator.__class__.__name__) for estimator, _, _
in estimators ]
    return ', '.join(result), {estimator_name.strip('"'): estimator for estima-
tor_name, (estimator, _, _) in zip(result, estimators)}
if __name__ == "__main__":
    estimators = get_best_estimators(True)
    estimators_str, estimator_dict = get_estimators_name(estimators)
    import argparse
    parser = argparse.ArgumentParser(description="""
        Testing emotion recognition system using your voice,
        please consider changing the model and/or parameters
as you wish.
        """)

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		25

```

parser.add_argument("-e", "--emotions", help=
                    """"Emotions to recognize separated by a comma
',', available emotions are
                    "neutral", "calm", "happy" "sad", "angry", "fear",
"disgust", "ps" (pleasant surprise)
                    and "boredom", default is "sad,neutral,happy"
                    """" , default="sad,neutral,happy")

parser.add_argument("-m", "--model", help=
                    """"
                    The model to use, 8 models available are: { },
                    default is "BaggingClassifier"
                    """" .format(estimators_str), default="BaggingClassi-
fier")

# Parse the arguments passed
args = parser.parse_args()
features = ["mfcc", "chroma", "mel"]
detector = EmotionRecognizer(estimator_dict[args.model], emo-
tions=args.emotions.split(","), features=features, verbose=0)

detector.train()
print("Test accuracy score: {:.3f}%".format(detector.test_score()*100))
print("Please talk")
filename = "test.wav"
record_to_file(filename)
result = detector.predict(filename)
print(result)

#Модуль для распознавания лица
from tensorflow.keras import Sequential
from tensorflow.keras.models import load_model
import cv2

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
						26
Изм.	Лист.	№ документа	Подп.	Дата.		

```

import numpy as np
from tensorflow.keras.preprocessing.image import img_to_array
# загружаем модель
model = Sequential()
classifier = load_model('ferjj.h5') # в нашей модели будет 6 классов для распознавания
# поэтому назначим 6 меток для нашей модели
class_labels = {0: 'Angry', 1: 'Fear', 2: 'Happy', 3: 'Neutral', 4: 'Sad', 5: 'Surprise'}
classes = list(class_labels.values())
# print(class_labels)
face_classifier = cv2.CascadeClassifier('./Haarcascades/haarcascade_frontalface_default.xml')
# эта функция используется для задания внешнего вида надписи, отображающей распознанную эмоцию
def text_on_detected_boxes(text,text_x,text_y,image,font_scale = 1,
                           font = cv2.FONT_HERSHEY_SIMPLEX,
                           FONT_COLOR = (0, 0, 0),
                           FONT_THICKNESS = 2,
                           rectangle_bgr = (0, 255, 0)):
    # определяем ширину и высоту текстового поля
    (text_width, text_height) = cv2.getTextSize(text, font, fontScale=font_scale,
    thickness=2)[0]
    # Set the Coordinates of the boxes
    box_coords = ((text_x-10, text_y+4), (text_x + text_width+10, text_y -
    text_height-5))
    # Draw the detected boxes and labels
    cv2.rectangle(image, box_coords[0], box_coords[1], rectangle_bgr,
    cv2.FILLED)

```

					<b>СКБФЭУ.1.ИП.01000000</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		27

```

cv2.putText(image, text, (text_x, text_y), font, fontScale=font_scale,
color=FONT_COLOR,thickness=FONT_THICKNESS)
# распознавание эмоций на изображении
def face_detector_image(img):
    gray = cv2.cvtColor(img.copy(), cv2.COLOR_BGR2GRAY) # конвертируем
изображение в серое
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    if faces is ():
        return (0, 0, 0, 0), np.zeros((48, 48), np.uint8), img
    allfaces = []
    rects = []
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_gray = cv2.resize(roi_gray, (48, 48), interpolation=cv2.INTER_AREA)
        allfaces.append(roi_gray)
        rects.append((x, w, y, h))
    return rects, allfaces, img
def emotionImage(imgPath):
    img = cv2.imread(imgPath)
    rects, faces, image = face_detector_image(img)
    i = 0
    for face in faces:
        roi = face.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)
        # определяем область, существенную для анализа (ROI), затем осуществ-
ляем поиск класса
        preds = classifier.predict(roi)[0]

```

```

label = class_labels[preds.argmax()]
label_position = (rects[i][0] + int((rects[i][1] / 2)), abs(rects[i][2] - 10))
i = + 1
# выводим название распознанной эмоции поверх изображения
text_on_detected_boxes(label, label_position[0],label_position[1], image)
cv2.imshow("Emotion Detector", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
# распознавание эмоций в видеопотоке
def face_detector_video(img):
    # преобразуем изображение в серое
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    if faces is ():
        return (0, 0, 0, 0), np.zeros((48, 48), np.uint8), img
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), thickness=2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_gray = cv2.resize(roi_gray, (48, 48), interpolation=cv2.INTER_AREA)
    return (x, w, y, h), roi_gray, img
def emotionVideo(cap):
    while True:
        ret, frame = cap.read()
        rect, face, image = face_detector_video(frame)
        if np.sum([face]) != 0.0:
            roi = face.astype("float") / 255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi, axis=0)

```

					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		29

```

# определяем область, существенную для анализа (ROI), затем осу-
ществляем поиск класса

preds = classifier.predict(roi)[0]
label = class_labels[preds.argmax()]
label_position = (rect[0] + rect[1]//50, rect[2] + rect[3]//50)
text_on_detected_boxes(label, label_position[0], label_position[1], image) #

```

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ЭТУ ФУНКЦИЮ В ДРУГИХ СВОИХ ПРОЕКТАХ НА ОСНОВЕ opencv

```

fps = cap.get(cv2.CAP_PROP_FPS)
cv2.putText(image, str(fps),(5, 40), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 255, 0), 2)
else:
    cv2.putText(image, "No Face Found", (5, 40), cv2.FONT_HERSHEY_SIM-
PLEX, 1, (255, 0, 0), 2)
cv2.imshow('All', image)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
if __name__ == '__main__':
    camera = cv2.VideoCapture(0) # если вы используете USB камеру, то исполь-
зуйте в качестве аргумента этой функции 1 вместо 0
emotionVideo(camera)
# IMAGE_PATH = "provide the image path"
# emotionImage(IMAGE_PATH) # если вы собираетесь распознавать эмоции
не в видеопотоке, а на изображении, то раскомментируйте эту строку

```

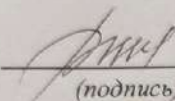
					<b>СКБФЭУ.1.ИП.01000000</b>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		30

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

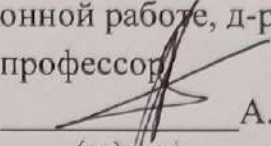
СОГЛАСОВАНО

Начальник отдела ОНИПКРС

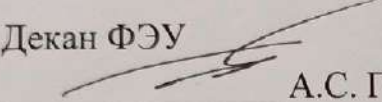
  
(подпись) Е.М. Димитриади  
« 23 » 01 20 24 г.

УТВЕРЖДАЮ

Проректор по науке и инновационной работе, д-р техн. наук.  
профессор

  
(подпись) А.В. Космынин  
« 23 » 01 20 24 г.

Декан ФЭУ

  
(подпись) А.С. Гудим  
« 23 » 01 20 24 г.

АКТ

о приемке в эксплуатацию проекта  
«EmoCruit: Эмоциональный интеллект в рекрутинге»

г. Комсомольск-на-Амуре

« 23 » 01 20 24 г.

Комиссия в составе представителей:

со стороны заказчика

- В.В. Солецкий – руководитель СКБ,
- А.С. Гудим – декан ФЭУ

со стороны исполнителя

- Р.В. Шибeko – руководителя проекта,

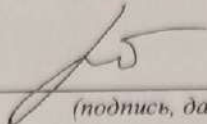
составила акт о нижеследующем:

«Исполнитель» передает проект «EmoCruit: Эмоциональный интеллект в рекрутинге», в составе:

1. Raspberry pi 4
2. Программный модуль

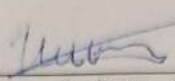
### 3. Руководство для пользователя

Руководитель проекта

  
\_\_\_\_\_  
(подпись, дата)

Р.В. Шибeko

Исполнители проекта

  
\_\_\_\_\_  
(подпись, дата)

Е.С. Ильченко