



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Колледж программирования и кибербезопасности

ДИПЛОМНЫЙ ПРОЕКТ

СПЕЦИАЛЬНОСТЬ 09.02.07

Информационные системы и программирование

на тему:

«Разработка приложения для конвертации
видеопотока в бегущую строку для распознавания
языка жестов»

Выполнил студент
группы ЩПКО-02-20 (ПКС-42)

_____ С.С. Сова
подпись ФИО студента

Руководитель
_____ М.Л. Мымрина
подпись ФИО руководителя

Нормоконтроль
_____ М.Л. Мымрина
подпись ФИО руководителя

Москва 2024



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

Колледж программирования и кибербезопасности

РЕЦЕНЗИЯ

на дипломный проект студента Совы Сергея Сергеевича

Специальность 09.02.07 Информационные системы и программирование

Тема дипломного проекта: «Разработка приложения для конвертации видеопотока в бегущую строку для распознавания языка жестов»

Объем пояснительной записки 39 листов

Объем графической части 1 лист

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТА

1. Актуальность темы

Дело в том, что в данный момент не хватает инструментов, которые могли бы обеспечить коммуникацию с людьми, использующими жестовый язык. Большинство существующих программ либо не поддерживают русский жестовый язык, либо имеют ограниченный функционал. Поэтому было решено создать программу, поддерживающую два языка жестов и возможность смены языка на лету. Это позволяет пользователю легко переключаться между языками и использовать программу в разнообразных ситуациях общения.

2. Заключение о степени соответствия выполненного дипломного проекта заданию

Предъявленный на рецензирование дипломный проект полностью соответствует полученному студентом заданию.

3. Оценка качества выполнения пояснительной записки и практической части

Пояснительная записка дипломного проекта выполнена на высоком техническом уровне, что является показателем освоения учебного материала и умения осуществлять поиск, анализ и оценку информации.

Выполнение практической части дипломного проекта свидетельствует о профессиональных компетенциях студента по специальности 09.02.07 Информационные системы и программирование.

Качество подбора и изложения материала дипломного проекта высокое.

4. Перечень положительных качеств дипломного проекта

Разработанная информационная система является проектом с возможностью практического применения. Программный функционал проекта обеспечивает гибкость, масштабируемость и эффективность в обработке данных и выполнении задач. Пользовательский интерфейс для работы с системой не перегружен и удобен для работы. Тестирование системы показало хорошую скорость и стабильность работы.

5. Основные недостатки дипломного проекта

Существенных недостатков дипломного проекта не выявлено.

6. Отзыв о проекте в целом, его реальность и актуальность

Тема дипломного проекта полностью раскрыта. Поставленные задачи выполнены в полном объеме. Актуальность темы дипломного проекта подтверждается потребностью предприятия в замене существующих методов взаимодействия с людьми, использующими жестовый язык, из-за устаревания технологий и необходимости соблюдения новых стандартов доступности. Разработанная автором автоматизированная система распознавания жестового языка представляет собой основу для дальнейшего развития и адаптации под конкретные потребности предприятия.

Дипломный проект заслуживает отличной оценки, а Сова Сергей Сергеевич присвоения квалификации «техник - программист».

Оценка теоретической части 5 (отлично)

Оценка практической части 5 (отлично)

Оценка электронной презентации 5 (отлично)

_____ 2024 г

Подпись рецензента _____

(Фамилия, имя, отчество, место работы, должность рецензента дипломного проекта)



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

Колледж программирования и кибербезопасности

О Т З Ы В

руководителя о дипломном проекте студента Совы Сергея Сергеевича
Специальность 09.02.07 Информационные системы и программирование

Тема дипломного проекта: «Разработка приложения для конвертации видеопотока в бегущую строку для распознавания языка жестов»

ХАРАКТЕРИСТИКА РАБОТЫ ДИПЛОМНИКА

- объем выполненной работы: Все части дипломного проекта, а так же его электронная презентация выполнены в полном объеме, в соответствии с заданием.

Дипломный проект состоит из нескольких разделов. В теоретических разделах проводится анализ предметной области и существующих разработок по теме проекта. В практических разделах проекта производится проектируются архитектура, пользовательский интерфейс и процесс обучения нейросети, разрабатывается алгоритм работы приложения.

- самостоятельность работы: при выполнении дипломного проекта студент продемонстрировал высокую степень самостоятельности, инициативность, отличные аналитические способности, умение систематизировать собранную информацию, а так же делать самостоятельные выводы, предложения и обобщения.

- отзыв о проекте в целом, его реальность, актуальность, использование современных технологий:

Дипломный проект выполнен на высоком техническом уровне и отвечает всем требованиям, предъявленным в задании на дипломное проектирование. Проведено обоснование выбора инструментальных средств, разработан интуитивно-понятный интерфейс. Тема проекта актуальна, приложение имеет социальную направленность.

- недостатки проекта: существенных недостатков нет.

Оценка теоретической части 5 (отлично)

Оценка практической части 5 (отлично)

Оценка электронной презентации 5 (отлично)

Общая оценка 5 (отлично)

30 мая 2024 г

Подпись руководителя

Мымрина М.Л., преподаватель высшей квалификационной категории колледжа программирования и кибербезопасности

Фамилия, имя, отчество, место работы, должность руководителя дипломного проекта

Утверждаю
Председатель ПЦК
«Программирование»
_____ Мырина М.Л.
подпись ФИО
«15» апреля 2024 г.

Председатель ПЦК
«Программирование»

Мымрина М.Л.

подпись

ФНО

«15» апреля 2024 г.

ЗАДАНИЕ

на дипломный проект

студенту Сове Сергею Сергеевичу группы ЩПКО-02-20

по специальности 09.02.07 Информационные системы и программирование

Тема : «Разработка приложения для конвертации видеопотока в бегущую строку для распознавания языка жестов»

Пояснительная записка

Введение

1 Исследовательский раздел

- 1.1 Анализ предметной области
- 1.2 Анализ существующих разработок
- 1.3 Постановка задач и определение требований к разрабатываемому приложению

2 Проектирование приложения

- 2.1 Проектирование архитектуры
- 2.2 Проектирование пользовательского интерфейса
- 2.3 Проектирование процесса обучения нейросети

3 Разработка приложения

- | | |
|-----|--------------------------|
| 3.1 | Выбор средств разработки |
| 3.2 | Обучение нейронной сети |
| 3.3 | Разработка приложения |
| 3.4 | Тестирование приложения |

Заключение

Список использованных источников

Графическая часть проекта / Приложения

Приложение к заданию на дипломный проект.

Разработать приложение для перевода жестового языка в текст и в аудиопоток с использованием технологий машинного обучения и компьютерного зрения.

Критерии разработки:

- наличие графического интерфейса;
- обработка видеопотока в реальном времени;
- обнаружение рук и отслеживание их движений;
- создание модели машинного обучения для классификации жестов;
- проигрывание аудиофайлов.

Основные функциональные требования:

- перевод русского и английского жестового языков в текст и в аудиопоток;
- захват жестов в реальном времени с помощью камеры;
- переключение между русским и английским языками.

Приложение для перевода жестового языка должно содержать систему обработки и анализа визуальных данных, которая состоит из модуля захвата изображения с камеры и модуля компьютерного зрения для распознавания и интерпретации жестов.

Задание на дипломный
проект выдал

«16» апреля 2024 г.

Подпись руководителя
проекта

Мырина М.Л.
ФИО руководителя
проекта

Задание на дипломный
проект получил

«16» апреля 2024 г.

Подпись студента-
исполнителя проекта

Сова С.С.
ФИО студента-
исполнителя проекта

Срок представления к защите дипломного проекта: до «16» мая 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1 Анализ предметной области.....	5
1.2 Анализ существующих разработок	6
1.3 Постановка задач и определение требований к разрабатываемому приложению	9
2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	10
2.1 Проектирование архитектуры.....	10
2.1.1 Общее описание	10
2.1.2 Компоненты системы.....	10
2.1.3 Взаимодействие компонентов	11
2.1.4 Принцип работы	11
2.1.5 Адаптивность и масштабируемость	12
2.1.6 Безопасность и конфиденциальность.....	12
2.2 Проектирование пользовательского интерфейса.....	13
2.3 Проектирование процесса обучения нейросети.....	15
2.3.1 Сбор и подготовка данных	15
2.3.2 Проектирование модели	16
2.3.3 Обучение и валидация модели.....	17
3 РАЗРАБОТКА ПРИЛОЖЕНИЯ	18
3.1 Выбор средств разработки	18
3.2 Обучение нейронной сети	19
3.2.1 Сбор данных	19
3.2.2 Предобработка данных	20
3.2.3 Обучение модели.....	21
3.2.4 Оценка точности.....	22

					ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ			
Изм.	Лист.	№ докум.	Подпись	Дата	Разработка приложения для конвертации видеопотока в бегущую строку для распознавания языка жестов	Лист	Лист	Листов
Разраб.		Сова С.С.				у	1	36
Руковод.		Мырина М.Л.						
Председ.ПЦК.		Мырина М.Л.						
					Пояснительная записка	КПК		

3.3 Разработка приложения	22
3.3.1 Инициализация и загрузка моделей	22
3.3.2 Инициализация MediaPipe.....	23
3.3.3 Словари жестов	23
3.3.4 Функции для переключения языка.....	24
3.3.5 Создание графического интерфейса	24
3.3.6 Обработка видеопотока и распознавание жестов	25
3.3.7 Воспроизведение аудиопотока	26
3.3.8 Обновление истории распознанных символов	26
3.3.9 Основной цикл приложения.....	27
3.4 Тестирование приложения	28
3.4.1 Тестирование на отказ и восстановление	28
3.4.2 Тестирование безопасности	29
3.4.3 Тестирование надёжности	30
3.4.4 Тестирование производительности	31
3.4.5 Тестирование удобства пользования.....	31
3.4.6 Нагрузочное тестирование	32
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	35

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	2

ВВЕДЕНИЕ

В окружающем мире, где технологии играют все более значительную роль в повседневной жизни, важность создания инклюзивного общества, доступного для всех, безусловно, растет. Особое внимание в этом контексте заслуживает общение и взаимопонимание между людьми, в том числе теми, кто использует язык жестов как основной способ коммуникации. Разработка программного обеспечения для распознавания языка жестов представляет собой шаг на пути к устранению барьеров между слышащими людьми и теми, кто имеет нарушения слуха или речи. Таким образом, актуальность и важность создания приложений, способных распознавать и переводить язык жестов, несомненна [1].

Целью данного дипломного проекта является разработка программного обеспечения, способного распознавать язык жестов в реальном времени и переводить его в письменную и устную форму, что обеспечивает беспрепятственное общение между людьми с нарушениями слуха и теми, кто не знаком с языком жестов.

Для достижения этой цели были поставлены следующие задачи разработки:

- изучение существующих методов распознавания жестов и их применение в разработке программного обеспечения;
- разработка алгоритма для распознавания языка жестов с использованием современных технологий машинного обучения и компьютерного зрения;
- создание пользовательского интерфейса, который был бы интуитивно понятен и удобен для использования целевой аудиторией;
- тестирование и оптимизация программного обеспечения для обеспечения его высокой точности и производительности.

Актуальность данного проекта обусловлена не только растущим спросом на инклюзивные технологии, но и специфическими потребностями

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				3

людей с нарушениями слуха, которые сталкиваются с трудностями в общении в повседневной жизни. Разработка такого приложения не только облегчит коммуникацию, но и поможет интегрировать людей с нарушениями слуха в общество, повысит их самостоятельность и качество жизни [2].

Практическое применение проекта обширно и включает в себя использование в образовательных учреждениях, на рабочих местах, в медицинских и социальных учреждениях, а также в повседневной жизни. Целевая аудитория проекта включает в себя не только людей с нарушениями слуха, но и тех, кто хочет научиться языку жестов, педагогов, социальных работников и всех, кто заинтересован в создании инклюзивного общества [2].

Таким образом, разработка программного обеспечения для распознавания языка жестов является актуальной и социально значимой задачей, решение которой способствует укреплению взаимопонимания и равноправия в обществе.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				4

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ предметной области

Язык жестов, или жестовая речь, является главным методом общения для людей с нарушением слуха или речи. Согласно последним данным Всемирной организации здравоохранения, 5% населения земного шара страдают нарушениями слуха. Это означает, что более четырехсот шестидесяти миллионов человек во всем мире страдают от потери слуха. Из них тридцать четыре миллиона детей. Кроме того, ожидается, что к 2050 году более девятисот миллионов человек будут страдать от потери слуха, а один миллиард молодых людей подвержены риску потери слуха из-за воздействия шума и других связанных с ним проблем. Нерешенная потеря слуха приводит к глобальным затратам в размере семидесяти триллионов рублей. Потеря слуха подразделяется на легкую, умеренную, тяжелую и глубокую категории в зависимости от интенсивности глухоты. Люди с тяжелой или глубокой потерей слуха не могут слушать других и, таким образом, сталкиваются с проблемами в общении. Этот недостаток общения может оказать сильное влияние на личность глухого человека, что может включать одиночество, изоляцию и разочарование [3].

Язык жестов — это особый тип языка, который используется глухими людьми в качестве способа общения. В отличие от других естественных языков, он использует осмысленные движения тела для передачи сообщений, и эти движения тела называются жестами или знаками. Движения рук и пальцев используются для передачи смысла. Язык жестов является полноценным естественным языком со своим синтаксисом и грамматикой. Разговорные языки варьируются от одного региона к другому, и в мире существует около семи тысяч разговорных языков. Точно так же не существует универсального жестового языка, и в мире существует около двухсот различных жестовых языков. Технологический прогресс современного мира и процесс глобализации также способствовал росту исследования языка жеста для цифрового перевода [3].

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				5

Интеграция людей с нарушениями слуха в общество немыслима без цифровизации языка жестов. Эта задача приобретает все большую актуальность в связи с бурным развитием технологий, которые открывают новые горизонты для обеспечения доступной и инклюзивной коммуникации. Цифровой перевод языка жестов облегчает взаимодействие между людьми с нарушениями слуха или речи и остальным обществом, делая его более доступным и эффективным. Такая инновация способствует устранению барьеров в коммуникации, сокращая зависимость от сурдопереводчиков и предоставляя возможность для независимого общения. Это не только повышает качество жизни людей с нарушениями слуха, но и способствует их полноценной социализации в обществе [3].

1.2 Анализ существующих разработок

В процессе подготовки данного проекта были рассмотрены и проанализированы следующие приложения: SLAIT, Mimix Sign Language Translator, Ace ASL.

Разрабатываемое приложение сравнивалось исключительно с зарубежными программами, так как российские аналоги в открытом доступе отсутствуют. Это выделяет разрабатываемый проект как уникальное решение, потенциально заполняющее пробел в доступности программ распознавания жестового языка для русскоязычных пользователей.

Первое приложение — SLAIT, имеет функцию распознавания жестов и речи, поддерживает только английский язык и способно переводить жесты в текст. Кроме того, оно предлагает дополнительную функцию перевода речи в текст (рисунок 1) [4].

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				6



Рис. 1. Фрагмент работы приложения «SLAIT»

Второе приложение — Mimix Sign Language Translator, также распознаёт жесты, но не имеет функции распознавания речи. Оно также поддерживает только английский язык и переводит жесты в текст. Уникальной особенностью является наличие 3D-аватара для демонстрации жестов (рисунок 2) [5].



Рис. 2. Фрагмент работы приложения «Mimix Sign Language Translator»

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				7

Третье приложение — Ace ASL, обладает возможностью распознавания жестов, но не распознаёт речь. Поддержка языка ограничена английским, приложение может переводить жесты в текст и содержит обучающий контент (рисунок 3) [6].

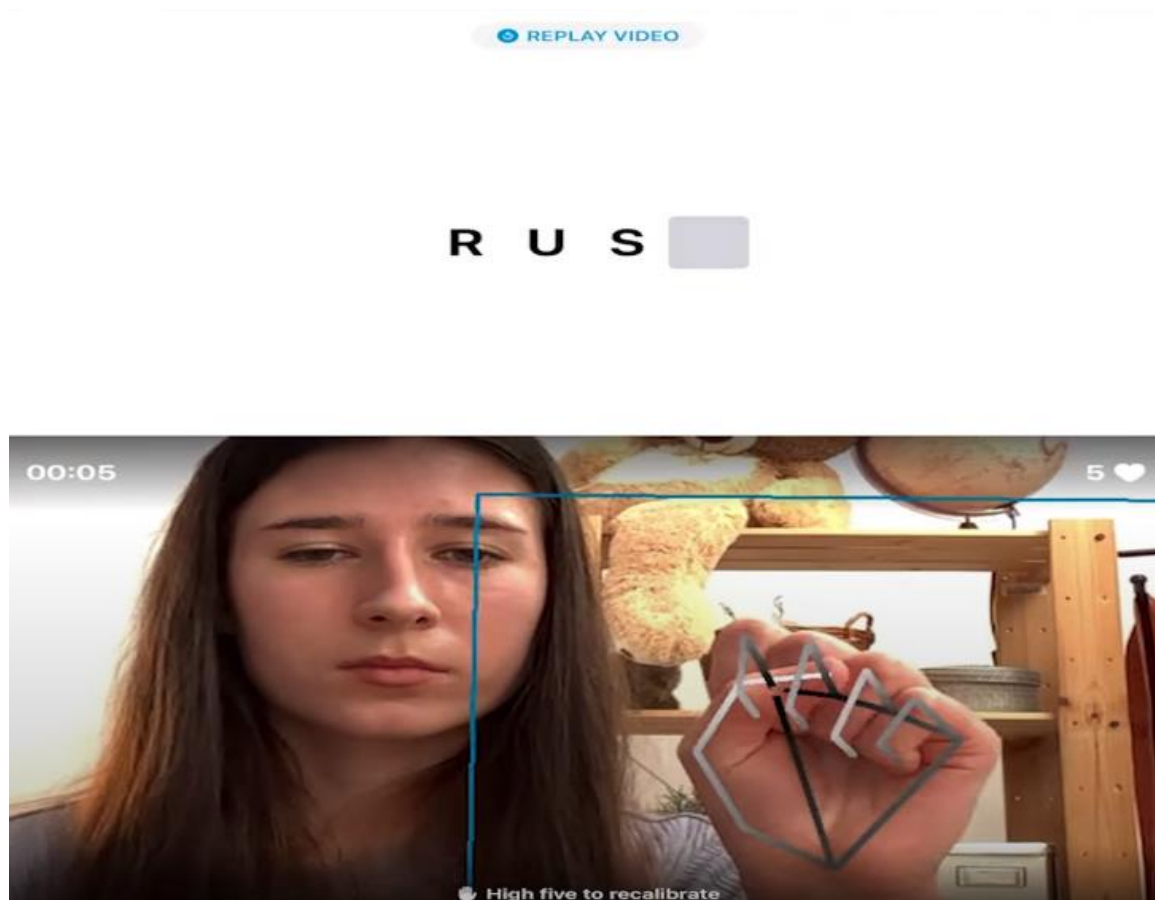


Рис. 3. Фрагмент работы приложения «Ace ASL»

Разрабатываемое приложение предоставляет те же основные функции, что и другие: распознавание жестов и перевод их в текст. Но, в отличие от аналогов, оно поддерживает не только английский язык, но и русский, и предлагает дополнительные функции, такие как озвучивание жеста, переключение языка и локальное хранение моделей.

На основании проведенного анализа можно сделать вывод, что разрабатываемая система имеет потенциал стать конкурентоспособным продуктом на рынке благодаря своим уникальным функциям. Включение поддержки русского языка выделяет её среди других аналогов, которые

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				8

ориентированы преимущественно на англоязычный сегмент. Дополнительные функции, такие как локальное хранение моделей и возможность работы в оффлайн-режиме, увеличивают ее привлекательность в регионах с нестабильным интернет-соединением или в условиях, где требуется высокая конфиденциальность данных. Таким образом, разрабатываемая система обладает всеми характеристиками, необходимыми для достижения успеха на рынке и оказания значительного социального влияния.

1.3 Постановка задач и определение требований к разрабатываемому приложению

Исходя из проведенного анализа, для разрабатываемой системы распознавания языка жестов должны быть соблюдены следующие требования:

- реализация распознавания языка жестов в реальном времени: система должна быть способна распознавать и интерпретировать язык жестов в реальном времени с использованием камеры;
- система должна поддерживать распознавание широкого спектра жестов;
- вывод результатов в текстовом и аудио-форматах: результаты распознавания жестов должны выводиться в текстовой форме с возможностью синтеза речи для облегчения восприятия информации людьми с нарушениями слуха;
- адаптивность интерфейса: интерфейс пользователя должен быть интуитивно понятным и адаптированным для использования различными категориями пользователей, включая людей с ограниченными возможностями;
- мультиязычность: система должна поддерживать распознавание русского и английского языков жестов, обеспечивая мультиязычную поддержку для широкого круга пользователей.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	9

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 Проектирование архитектуры

2.1.1 Общее описание

Архитектура программы является ключевым аспектом разработки, определяющим её гибкость, масштабируемость и легкость в поддержке. Проектирование архитектуры предусматривает создание чёткой структуры проекта, разделение его на компоненты и определение взаимодействий между ними.

Проект системы распознавания языка жестов представляет собой комплексное решение, объединяющее алгоритмы компьютерного зрения, машинное обучение и аудиовизуальные средства для идентификации и интерпретации языка жестов в реальном времени. Система разработана для обеспечения возможности общения с глухими и слабослышащими людьми, переводя жесты в аудиоформат и текст.

2.1.2 Компоненты системы

Модуль захвата изображения отвечает за получение видеопотока с камеры и захват отдельных кадров для последующей обработки. Этот модуль является входной точкой данных в систему.

Модуль предобработки изображений реализует алгоритмы предварительной обработки изображений, такие как фильтрация, нормализация и преобразование изображений, чтобы улучшить качество и эффективность распознавания.

Модуль обнаружения жестов содержит алгоритмы компьютерного зрения и машинного обучения для обнаружения и классификации жестов на изображениях. Этот модуль использует обученные модели для определения конкретных жестов языка жестов.

Модуль воспроизведения звука отвечает за воспроизведение звуковой интерпретации распознанных жестов. Модуль поддерживает добавление новых звуков и легко настраивается под различные языки.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	10

Пользовательский интерфейс предоставляет графическую оболочку для взаимодействия с пользователем, отображает видеопоток и результаты распознавания жестов в реальном времени, а также позволяет настраивать параметры системы.

2.1.3 Взаимодействие компонентов

Видеопоток поступает из модуля захвата изображения в модуль предобработки, где изображение подготавливается для анализа. Далее обработанные данные передаются в модуль обнаружения жестов, где с помощью алгоритмов машинного обучения происходит классификация жестов. Результаты классификации отправляются в модуль воспроизведения звука для аудиовыхода и одновременно отображаются в пользовательском интерфейсе.

Пользовательский интерфейс позволяет выбрать нужный язык жестов. Изменения обрабатываются в соответствующих модулях для адаптации процесса распознавания.

2.1.4 Принцип работы

Принцип работы разрабатываемой системы распознавания языка жестов состоит из следующих этапов:

- 1) пользователь запускает систему и вступает в визуальный контакт с веб-камерой;
- 2) система захватывает видеопоток, из которого извлекает кадры для анализа;
- 3) кадры обрабатываются для выделения рук и их ориентации;
- 4) извлеченные данные подаются на вход модели машинного обучения, которая классифицирует жест;
- 5) после успешного распознавания жеста система воспроизводит соответствующий аудиофайл и отображает текстовую интерпретацию жеста;

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	11

б) пользователь получает аудио- и визуальную обратную связь с помощью графического интерфейса пользователя, позволяющую понять, какой жест был распознан системой. Это обеспечивает мгновенную реакцию на действия пользователя и способствует эффективному общению между пользователями и системой.

Также была разработана блок-схема принципа работы приложения. Блок-схема создана с помощью веб-инструмента «Draw.io».

Draw.io — это сервис, который позволяет пользователям создавать различные типы диаграмм, такие как блок-схемы, организационные схемы, UML диаграммы, ER диаграммы и многое другое. Он интуитивно понятен в использовании и поддерживает совместную работу в реальном времени, что позволяет нескольким пользователям одновременно работать над одной и той же диаграммой. Draw.io интегрирован с различными облачными хранилищами, что позволяет легко сохранять и делиться диаграммами.

Блок-схема представлена в графической части.

2.1.5 Адаптивность и масштабируемость

Система спроектирована с возможностью изменения для различных языков жестов и расширения функционала. Благодаря модульной архитектуре, в систему можно интегрировать дополнительные модели машинного обучения для распознавания большего количества жестов или поддержки новых языков жестов. Также предусмотрена возможность обновления и добавления аудиофайлов для соответствующих жестов, что делает систему гибкой и масштабируемой.

2.1.6 Безопасность и конфиденциальность

В проектировании системы уделено внимание обеспечению безопасности и конфиденциальности пользовательских данных. Все данные, собранные во время работы с системой, обрабатываются локально, без передачи на внешние серверы. Это гарантирует защиту личной информации пользователей и их конфиденциальность.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	12

2.2 Проектирование пользовательского интерфейса.

Проектирование дизайна пользовательского интерфейса для программы распознавания языка жестов включает в себя несколько ключевых аспектов, которые должны быть тщательно продуманы для обеспечения удобства и эффективности использования приложения.

Основными компонентами интерфейса являются окно приложения, панель управления с кнопками переключения языка, индикатором текущего языка, отображение видеопотока, отображение распознанных жестов с текстовым выводом и воспроизведение звука для аудиальной обратной связи при распознавании жестов.

Окно приложения служит контейнером для всех элементов управления и отображаемых данных. Оно должно быть достаточно просторным для комфортного размещения всех компонентов интерфейса, а также обеспечивать хорошую видимость распознаваемых жестов.

Панель управления включает в себя кнопки переключения языка и индикатор текущего языка. Кнопки переключения языка позволяют пользователю выбрать язык жестов, на котором будет осуществляться распознавание. Индикатор текущего языка визуализирует выбранный язык жестов.

В области отображения видеопотока в реальном времени показывается видеопоток с камеры. Важно обеспечить качественное отображение изображения, чтобы пользователь мог точно следить за своими жестами.

Распознанные жесты отображаются в текстовом виде. Текст должен быть крупным и читаемым, чтобы пользователь мог легко его воспринимать.

В соответствии распознанному жесту воспроизводится звуковой файл.

Принципами дизайна являются интуитивность, доступность, отзывчивость и эстетика:

- интерфейс должен быть понятным с первого взгляда;
- элементы управления должны быть легкодоступными и удобными для использования людьми с различными физическими возможностями;

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	13

- интерфейс должен быстро реагировать на действия пользователя, обеспечивая мгновенную обратную связь;
- дизайн должен быть приятным и современным, способствующим длительной работе с программой.

Макет окна был создан с помощью wireframe.cc – приложения для быстрого и комфортного проектирования интерфейсов [7].

Его основные особенности:

- минималистичный интерфейс – небольшая панель кнопок и окно устройства;
- шаблоны окон в альбомном и портретном исполнении;
- удобный способ добавления элементов: drag-to-draw и только потом выбор типа объекта;
- легко добавляются комментарии к интерфейсу;
- простая цветовая палитра;
- горячие клавиши: стандартные Ctrl+C / Ctrl+V, Alt + перемещение;
- указателя манипулятора типа «мышь» для дублирования объекта и двойной клик для редактирования.

Макет экрана представлен на рисунке 4.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				14

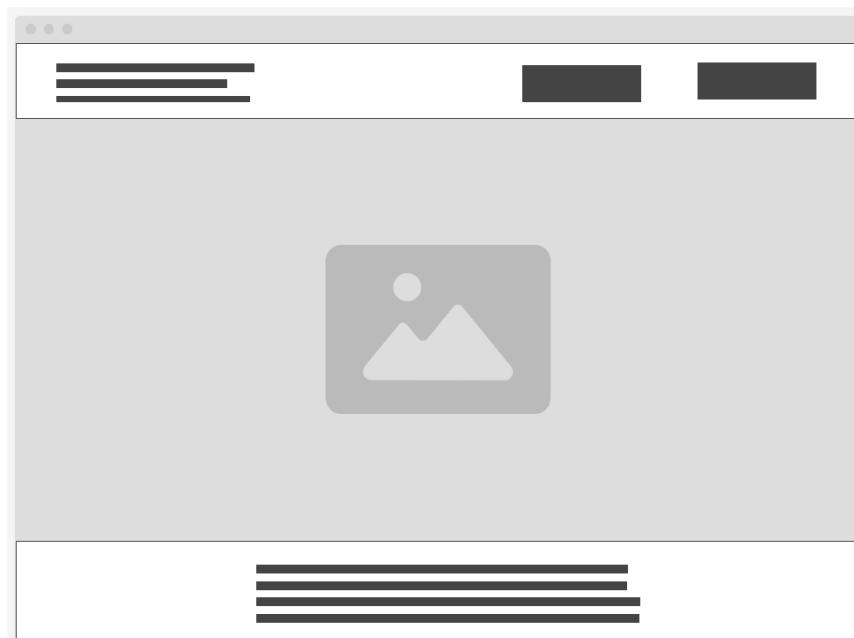


Рис. 4 Макет экрана

2.3 Проектирование процесса обучения нейросети

2.3.1 Сбор и подготовка данных

Проектирование процесса обучения нейросети для распознавания языка жестов является многоэтапной задачей, включающей сбор и подготовку данных, проектирование архитектуры модели, обучение и тестирование.

Сбор и подготовка данных являются критически важными этапами в процессе обучения нейросетей, особенно когда речь идет о распознавании изображений, таких как жесты. Эти этапы включают в себя ряд ключевых действий, начиная от первоначального сбора данных до их последующей обработки для улучшения качества и эффективности обучения нейросети.

Сбор данных – это процесс собирания информации из различных источников для последующего использования в аналитических или обучающих целях. В контексте обучения нейросетей под данными часто понимаются изображения, видеопотоки, тексты или звуковые файлы, которые могут быть использованы для обучения модели выполнять определенную задачу.

В данном случае, данные представляют собой изображения жестов. Эти

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				15

изображения собраны с помощью веб-камеры. Очень важно обеспечить разнообразие собираемых данных, что включает в себя получение изображений жестов в различных условиях освещения, с разными фонами и в разнообразных позициях рук. Это помогает улучшить способность нейросети корректно распознавать жесты в различных ситуациях.

Подготовка данных – это процесс очистки и трансформации сырых данных в формат, подходящий для обучения моделей. Включает в себя такие шаги, как нормализация, масштабирование, увеличение объема данных и другие.

Нормализация подразумевает приведение всех числовых характеристик к единому масштабу, например, изменение разрешения изображений так, чтобы они лежали в одном диапазоне значений. Это уменьшает предвзятость модели по отношению к определенным особенностям данных.

Масштабирование обеспечивает, чтобы все изображения имели одинаковый размер, что необходимо для обработки нейросетью.

Преобразование изображений может включать изменение угла наклона изображения, его поворот или отражение. Это позволяет увеличить разнообразие обучающего набора данных без необходимости собирать дополнительные изображения.

Данные шаги помогают повысить качество распознавания и уменьшить вероятность переобучения модели на конкретных особенностях тренировочного набора данных.

2.3.2 Проектирование модели

Следующий шаг - проектирование архитектуры нейронной сети, которая будет использоваться для распознавания жестов. Модель базируется на сверточных нейронных сетях, которые хорошо подходят для задач обработки изображений благодаря их способности выявлять важные признаки на различных уровнях абстракции.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	16

Выбор архитектуры модели зависит от многих факторов, включая сложность задачи, доступные вычислительные ресурсы и требуемую точность распознавания. Важно найти баланс между глубиной и шириной сети, чтобы обеспечить достаточную выразительную способность при сохранении умеренных требований к вычислительным ресурсам.

2.3.3 Обучение и валидация модели

На этапе после разработки структуры модели наступает время для её обучения с использованием заранее подготовленного наборов данных. Этот процесс включает в себя адаптацию параметров модели, таких как веса, через методы оптимизации для минимизации значения функции потерь. Функция потерь вычисляет разницу между выходными данными модели и фактическими значениями, представляя степень ошибки предсказаний модели. Сущность обучения заключается в том, чтобы корректировать веса модели таким образом, чтобы она могла точно предсказывать истинные значения, минимизируя ошибку предсказания.

В процессе обучения применяется несколько эпох, в каждой из которых модель проходит через весь набор данных, пытаясь улучшить свои предсказательные способности. После каждой эпохи можно оценить качество обучения модели, используя различные метрики, такие как точность, потери и др.

Чтобы предотвратить переобучение, когда модель хорошо работает на обучающих данных, но плохо адаптируется к новым данным, используется метод валидации. Для этого отдельный валидационный набор данных, который не использовался в процессе обучения, применяется для проверки эффективности модели. Валидация помогает убедиться, что модель не только запоминает обучающий набор, но и обладает способностью к обобщению, то есть эффективно работает с новыми данными, которые не встречались в процессе обучения. Этот шаг критически важен для подтверждения универсальности и надежности модели перед её практическим применением.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				17

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

3.1 Выбор средств разработки

При разработке программы для распознавания языка жестов использовался язык программирования Python и библиотеки OpenCV, MediaPipe, Scikit-learn, Pickle, Pillow, Tkinter и Pygame.

Python является языком программирования общего назначения, хорошо подходящий для работы с машинным обучением и обработкой данных благодаря широкому набору библиотек и фреймворков [8].

OpenCV является одна из основных библиотек для работы с изображениями и видеопотоками в проектах, связанных с компьютерным зрением. В данном проекте используется для захвата видеопотока с камеры и предварительной обработки изображений перед передачей их в модель распознавания [9].

Библиотека MediaPipe от Google предназначена для обработки медиафайлов, включая анализ видеопотоков и изображений. В проекте используется для определения положения рук на изображениях с помощью предварительно обученной модели распознавания рук [10].

Библиотека для машинного обучения Scikit-learn, предоставляющая различные алгоритмы классификации, регрессии и кластеризации. Используется для обучения модели классификации (RandomForestClassifier) на основе обработанных данных для распознавания жестов [11].

Pickle — это модуль для сериализации и десериализации структуры объектов Python. Применяется для сохранения и загрузки обученной модели машинного обучения [12].

Pillow и Tkinter используются для создания графического интерфейса пользователя и отображения обработанных изображений. PIL позволяет обрабатывать изображения в Python [13], а Tkinter - создавать оконные приложения [14].

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				18

Pygame является библиотекой для создания видеоигр, которая в данном случае используется для воспроизведения звуков при распознавании определенных жестов [15].

Выбор данных инструментов обусловлен их широкими возможностями в области обработки изображений, машинного обучения и разработки пользовательских интерфейсов, а также поддержкой большого сообщества, что облегчает решение возникающих проблем и вопросов в процессе разработки.

3.2 Обучение нейронной сети

3.2.1 Сбор данных

Для сбора данных и создания обучающего набора данных использовались библиотеки OpenCV для работы с видеопотоком и MediaPipe для распознавания ключевых точек (landmarks) на изображениях рук. Для каждого жеста было создано 150 изображений, что обеспечило достаточный объем данных для обучения модели. Сбор данных проводился в разнообразных условиях для повышения устойчивости и надёжности модели. Процесс включает захват изображений каждого жеста и запись соответствующих ключевых точек в базу данных.

```
import cv2
import mediapipe as mp
mp_hands = mp.solutions.hands
hands=mp_hands.Hands(static_image_mode=True,min_detection_confidence=0.3)
cap = cv2.VideoCapture(0)
for class_id in range(number_of_classes):
    class_dir = os.path.join(DATA_DIR, str(class_id))
    os.makedirs(class_dir, exist_ok=True)
    for i in range(dataset_size):
        ret, frame = cap.read()
        results=hands.process(cv2.cvtColor(frame,cv2.COLOR_BGR2RGB))
        if results.multi_hand_landmarks:
```

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПК0-02-20.18.06.24.ПЗ	19

```

for hand_landmarks in results.multi_hand_landmarks:
data = []
for landmark in hand_landmarks.landmark:
data.extend([landmark.x, landmark.y])
img_path = os.path.join(class_dir, f"{i}.jpg")
cv2.imwrite(img_path, frame)
data_list.append(data)
labels.append(class_id)

```

Для каждого класса жеста создается отдельная директория, в которую сохраняются изображения с распознанными жестами данного класса. Одновременно извлекаются ключевые точки рук и сохраняются в список «data_list» для дальнейшего формирования обучающего набора данных.

3.2.2 Предобработка данных

После сбора данных они преобразуются в формат, пригодный для обучения модели.

```

import pickle
data = []
labels = []
for class_id in range(number_of_classes):
class_dir = os.path.join(DATA_DIR, str(class_id))
for img_path in os.listdir(class_dir):
img = cv2.imread(os.path.join(class_dir, img_path))
results=hands.process(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
if results.multi_hand_landmarks:
for hand_landmarks in results.multi_hand_landmarks:
data_aux = []
for landmark in hand_landmarks.landmark:
data_aux.extend([(landmark.x - min_x) / (max_x - min_x),
(landmark.y - min_y) / (max_y - min_y)])
data.append(data_aux)
labels.append(class_id)
data = np.array(data)
labels = np.array(labels)

```

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	20

```
pickle.dump({'data':data,'labels':labels},open('data.pickle'
, 'wb'))
```

Скрипт загружает изображения из соответствующих директорий, извлекает ключевые точки рук и нормализует их значения относительно минимальных и максимальных значений для данного изображения.

Нормализация помогает улучшить обобщающую способность модели. Затем данные сохраняются в файл «data.pickle» для последующего использования в процессе обучения.

3.2.3 Обучение модели

Для обучения нейронной сети на задаче распознавания языка жестов был использован ансамбль решающих деревьев (Random Forest Classifier), обеспечивающий высокую точность и эффективность в задачах классификации. Данный метод выбран за его способность обрабатывать большие объемы данных с высокой степенью точности и устойчивости к переобучению.

Обучение модели происходит в скрипте с использованием библиотеки Scikit-learn. Данные были разделены на обучающую и тестовую выборки в соотношении 80% на 20%, что является стандартной практикой для обучения и оценки моделей машинного обучения. Это разделение помогает проверить, насколько хорошо модель может обобщать информацию на данных, которые она ранее не видела. В процессе обучения модель «Random Forest Classifier» адаптировала свои параметры таким образом, чтобы максимизировать точность классификации жестов.

```
import pickle

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
data_dict = pickle.load(open('data.pickle', 'rb'))
data = data_dict['data']
labels = data_dict['labels']

X_train, X_test, y_train, y_test = train_test_split(data,
labels, test_size=0.2, random_state=42)
```

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	21

```

model=RandomForestClassifier(n_estimators=100,random_state=4
2)
model.fit(X_train, y_train)

```

Данные загружаются из файла «data.pickle», затем разделяются на обучающую и тестовую выборки с помощью функции «train_test_split». Обучение модели «Random Forest Classifier» происходит на обучающей выборке с использованием метода «fit». Для оценки эффективности модели используется метрика «accuracy score».

```

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
pickle.dump({'model': model}, open('model.p', 'wb'))

```

Обученная модель сохраняется в файл «model.p» для последующего использования в приложении.

3.2.4 Оценка точности

Точность обученной модели оценивается на тестовой выборке. Для этого предсказания модели на тестовых данных сравниваются с реальными метками классов с помощью метрики «accuracy_score» из библиотеки Scikit-learn.

```

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

```

Полученная точность отражает долю корректно классифицированных примеров в тестовом наборе данных. В ходе экспериментов была достигнута точность около 90%, что свидетельствует о высокой эффективности обученной модели в распознавании жестов.

3.3 Разработка приложения

3.3.1 Инициализация и загрузка моделей

Загружаются предварительно обученные модели для распознавания русского и английского языков жестов из файлов «RuModel.p» и «EnModel.p»

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	22

с помощью модуля «pickle». Загруженные модели сохраняются в переменных «model_rus» и «model_eng» соответственно. Переменная «current_model» инициализируется моделью для русского языка жестов.

```
model_dict_rus = pickle.load(open('RuModel.p', 'rb'))
model_rus = model_dict_rus['model']
model_dict_eng = pickle.load(open('EnModel.p', 'rb'))
model_eng = model_dict_eng['model']
current_model = model_rus
```

3.3.2 Инициализация MediaPipe

Инициализируется модуль «hands» из библиотеки MediaPipe. Параметр «static_image_mode=True» указывает, что модель будет работать с отдельными изображениями. Параметр «min_detection_confidence=0.3» задает минимальный порог уверенности для распознавания жестов.

```
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
hands = mp_hands.Hands(static_image_mode=True,
min_detection_confidence=0.3)
```

3.3.3 Словари жестов

Словари связывают индексы, выдаваемые моделью, с соответствующими символами для русского и английского языков жестов. В данном случае, модель распознает цифры от 0 до 5 и буквы от А до Д (или от А до Е для английского языка) [16] [17].

```
labels_dict_rus = {0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5:
'5', 6: 'А', 7: 'Б', 8: 'В', 9: 'Г', 10: 'Д'}
labels_dict_eng = {0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5:
'5', 6: 'A', 7: 'B', 8: 'C', 9: 'D', 10: 'E'}
labels_dict = labels_dict_rus
```

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	23

3.3.4 Функции для переключения языка

Функция «switch_labels» изменяет используемый словарь «labels_dict» в зависимости от выбранного языка («labels_dict_rus» для русского, «labels_dict_eng» для английского). Функция «switch_language» изменяет используемую модель «current_model» и вызывает «switch_labels» для обновления словаря.

```
def switch_labels(lang):
    global labels_dict
    labels_dict = labels_dict_rus if lang == 'Rus' else
labels_dict_eng
def switch_language(lang):
    global current_model
    current_model = model_rus if lang == 'Rus' else model_eng
    switch_labels(lang)
```

3.3.5 Создание графического интерфейса

В этом фрагменте кода создается главное окно приложения с помощью библиотеки Tkinter. Создаются различные элементы интерфейса, такие как фреймы, метки и кнопки. Метка «label» используется для отображения видеопотока с камеры, «output_label» для вывода распознанных символов, а «class_label» для отображения истории распознанных символов.

```
window = tk.Tk()
window.title("Sign Language Translator")
label_frame = tk.Frame(window)
label_frame.pack(fill=tk.BOTH, expand=True)
label = tk.Label(label_frame)
label.pack(expand=True, padx=10, pady=10)
output_label = tk.Label(window, text='', font=('Arial', 12))
output_label.pack(side=tk.BOTTOM, fill=tk.X)
class_label = tk.Label(window, text='', font=('Arial', 12))
class_label.pack(side=tk.TOP, fill=tk.X)
```

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				24

3.3.6 Обработка видеопотока и распознавание жестов

Функция «update_frame» является ядром всего приложения. Она захватывает кадры с веб-камеры, обрабатывает их с помощью MediaPipe для распознавания жестов, а затем отображает результат в графическом интерфейсе.

Описание того, что происходит в этой функции:

- 1) считывается кадр с веб-камеры с помощью «cap.read()»;
- 2) кадр преобразуется в формат RGB с помощью «cv2.cvtColor»;
- 3) обработка кадра производится с помощью MediaPipe: «hands.process(frame_rgb)»;
- 4) если на кадре обнаружена рука, извлекаются координаты ключевых точек;
- 5) координаты ключевых точек преобразуются в форму, подходящую для модели;
- 6) модель «current_model» выполняет предсказание на основе преобразованных координат ключевых точек;
- 7) полученный индекс символа преобразуется в соответствующий символ с помощью словаря «labels_dict»;
- 8) если распознанный символ отличается от предыдущего, воспроизводится соответствующий звуковой файл с помощью «play_sound», обновляется метка «output_label» и вызывается функция «update_class_history» для обновления истории распознанных символов;
- 9) кадр преобразуется обратно в формат BGR и отображается в метке «label»;
- 10) функция «update_frame» вызывается снова через 10 миллисекунд с помощью «window.after(10, update_frame)».

Исходный код обработки видеопотока и распознавания жестов представлен в приложении 1.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	25

3.3.7 Воспроизведение аудиопотока

Функция «play_sound» используется для воспроизведения звуковых файлов, соответствующих распознанным символам. Она принимает имя распознанного символа («class_name») в качестве аргумента.

Внутри функции определяется префикс пути к звуковым файлам («sounds_path») на основе выбранного языка («RU» для русского или «EN» для английского). Затем формируется полный путь к соответствующему звуковому файлу в формате MP3 с помощью этого префикса и имени символа.

Далее с помощью библиотеки Pygame загружается и воспроизводится звуковой файл.

Переменная «last_class_name» используется для предотвращения повторного воспроизведения звука для одного и того же символа.

```
def play_sound(class_name):
    global last_class_name
    if class_name != last_class_name:
        lang_prefix = 'RU' if current_language.get() == 'Русский язык
        жестов' else 'EN'
        sound_file = f"{sounds_path[lang_prefix]}{class_name}.mp3"
        try:
            pygame.mixer.music.load(sound_file)
            pygame.mixer.music.play()
        except Exception as e:
            print(f"Error playing sound: {e}")
        last_class_name = class_name
```

3.3.8 Обновление истории распознанных символов

Функция «update_class_history» обновляет историю распознанных символов и отображает ее в метке «class_label».

Если распознанный символ («class_name») не является специальным символом «---» и отличается от последнего распознанного символа в истории, он добавляется в конец списка «class_history».

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	26

Если длина списка «class_history» превышает 5 элементов, удаляется самый старый элемент из начала списка с помощью «class_history.pop(0)».

Затем содержимое списка «class_history» объединяется в строку с разделителем «>» и устанавливается в качестве текста метки «class_label».

```
def update_class_history(class_name):
    global class_history
    if class_name != "---" and (not class_history or class_name
    != class_history[-1]):
        class_history.append(class_name)
        if len(class_history) > 5:
            class_history.pop(0)
        class_label.config(text=" > ".join(class_history))
```

3.3.9 Основной цикл приложения

В конце кода вызывается функция «update_frame», которая запускает цикл обработки видеопотока и распознавания жестов.

Затем с помощью «window.mainloop()» запускается главный цикл событий Tkinter, который обеспечивает отображение графического интерфейса и обработку событий пользователя.

После завершения работы приложения вызывается «cap.release()» для освобождения ресурсов, связанных с веб-камерой.

```
update_frame()
window.mainloop()
cap.release()
```

Таким образом, данное приложение использует несколько библиотек и технологий для распознавания языка жестов в режиме реального времени с помощью веб-камеры. Оно обеспечивает графический интерфейс для выбора языка (русский или английский), отображает распознанные символы, историю распознанных символов и воспроизводит соответствующие звуковые файлы.

Демонстрация работы приложения представлена на рисунке 5.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПК0-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				27

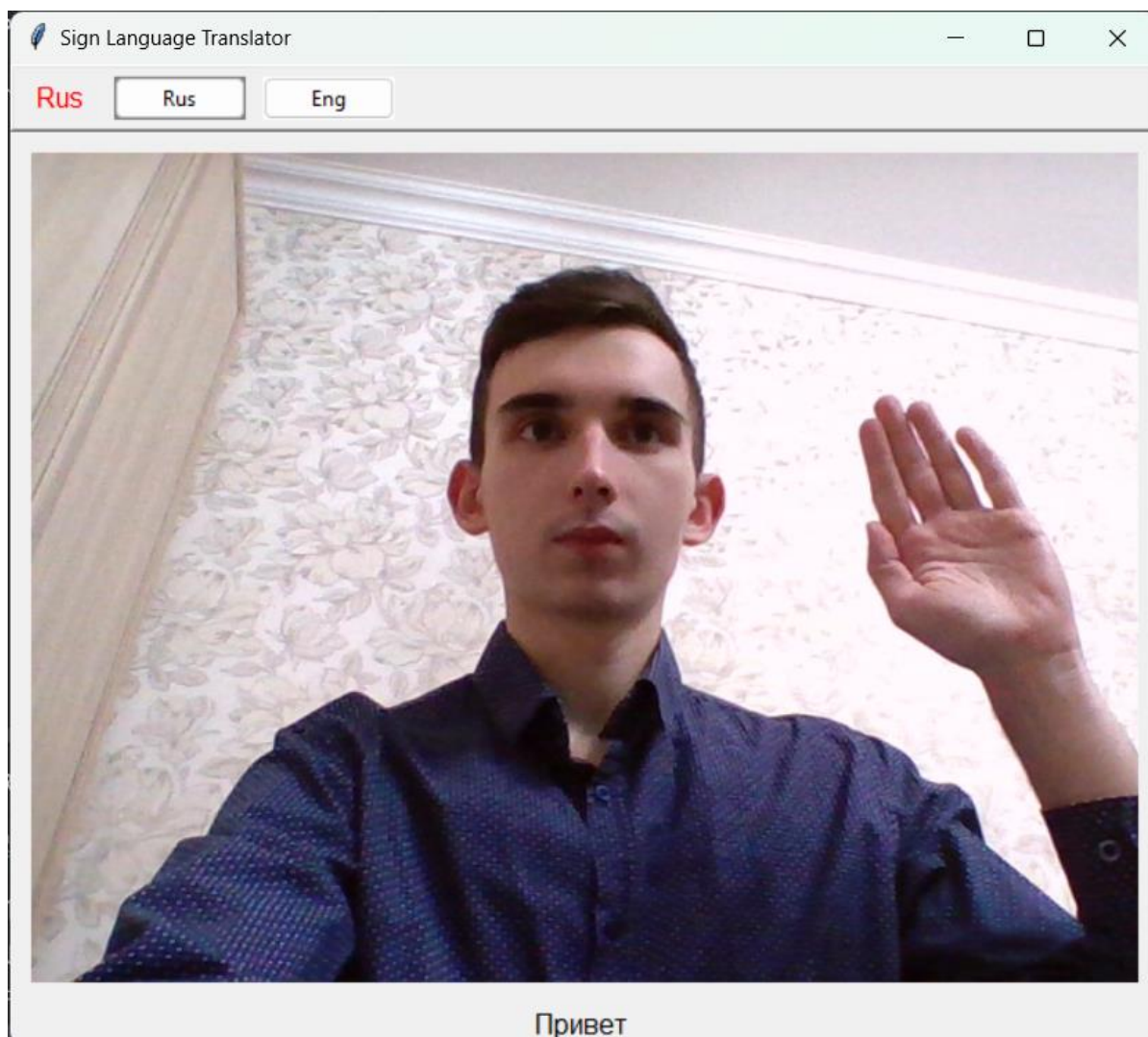


Рис. 5 Демонстрация работы приложения

3.4 Тестирование приложения

3.4.1 Тестирование на отказ и восстановление

Тестирование на отказ и восстановление — это процесс проверки способности системы или приложения справляться с потенциальными ошибками и восстанавливаться после сбоев.

Тест-кейс тестирования на отказ и восстановление представлен в таблице № 1.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				28

Тест-кейс для тестирования на отказ и восстановление

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на внезапное отключение питания	Внезапно отключить питание компьютера во время работы программы.	После восстановления питания и перезапуска программа продолжает работу нормально.	После восстановления питания и перезапуска программа продолжает работу нормально.
Тест на отказ веб-камеры	Отключить веб-камеру во время работы программы.	Программа корректно обрабатывает отключение и сообщает об ошибке.	Программа корректно обрабатывает отключение и сообщает об ошибке.
Тест на переполнение памяти	Искусственно создать условия для переполнения оперативной памяти компьютера, открыв большое количество вкладок в браузере.	Программа управляет нехваткой памяти без сбоев, предотвращая утечки памяти.	Программа управляет нехваткой памяти без сбоев, предотвращая утечки памяти.
Тест на восстановление после сбоя программы	Искусственно вызвать сбой программы через диспетчера задач.	Программа быстро восстанавливается после перезапуска, возвращаясь к стабильной работе.	Программа быстро восстанавливается после перезапуска, возвращаясь к стабильной работе.

3.4.2 Тестирование безопасности

Тестирование на безопасность нацелено на обнаружение уязвимостей, которые могут быть использованы для несанкционированного доступа, манипуляции данными или иных вредоносных действий.

Тест-кейс тестирования безопасности представлен в таблице № 2.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	29

Тест-кейс для тестирования безопасности

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на утечку информации	Анализировать сообщения об ошибках и логи на предмет утечки конфиденциальной информации.	Сообщения об ошибках и логи не содержат конфиденциальной информации, которая может быть использована для атаки.	Сообщения об ошибках и логи не содержат конфиденциальной информации, которая может быть использована для атаки.
Тест на неправильную настройку безопасности	Проверить конфигурацию и настройки безопасности программы. Перейти в раздел «Установочные пакеты» и убедиться, что установлены последние обновления безопасности	Конфигурации и настройки безопасности оптимизированы для защиты от известных уязвимостей.	Конфигурации и настройки безопасности оптимизированы для защиты от известных уязвимостей.

3.4.3 Тестирование надёжности

Целью тестирования надёжности является проверка способности системы работать без сбоев в различных условиях и на протяжении длительного периода времени. Это включает в себя проверку способности программы справляться с ошибками, восстанавливаться после сбоев и обрабатывать исключительные ситуации без потери данных или функциональности.

Тест-кейс тестирования надёжности представлен в таблице № 3.

Таблица № 3

Тест-кейс тестирования на надёжность

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на длительное исполнение	Запустить программу на 1 час.	Программа продолжает стабильно работать без утечек памяти и сбоев.	Программа продолжает стабильно работать без утечек памяти и сбоев.
Тест на корректную обработку внешних зависимостей	Отключить или изменить условия работы библиотек OpenCV, MediaPipe, Pygame.	Программа корректно обрабатывает сбои или изменения в работе внешних зависимостей.	Программа корректно обрабатывает сбои или изменения в работе внешних зависимостей.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				30

3.4.4 Тестирование производительности

Тестирование направлено на оценку скорости и эффективности работы программы в различных условиях. Это включает в себя тестирование времени реакции, скорости обработки данных, использования системных ресурсов и способности обрабатывать высокие нагрузки.

Тест-кейс тестирования производительности представлен в таблице №4.

Таблица № 4

Тест-кейс тестирования на производительность

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на время запуска программы	Замерить время от запуска программы до готовности к использованию.	Программа запускается и становится готовой к использованию через 30 секунд.	Программа запускается и становится готовой к использованию через 28 секунд.
Тест на скорость распознавания жестов	Замерить время от момента показа жеста до отображения результата на экране.	Результаты распознавания жестов отображаются в течение 1 секунды.	Результаты распознавания жестов отображаются в течение 0,5 секунд.
Тест на способность к многозадачности	Запустить программу параллельно с другими приложениями, создающими нагрузку на систему, например, с любыми видеоредакторами, играми или программами для 3D-моделирования	Программа продолжает эффективно работать, не демонстрируя значительного снижения производительности.	Программа продолжает эффективно работать, не демонстрируя значительного снижения производительности.

3.4.5 Тестирование удобства пользования

Тестирование удобства направлено на оценку того, насколько легко и интуитивно понятно конечным пользователям, как взаимодействовать с приложением. Это включает в себя анализ лёгкости освоения программы новыми пользователями, эффективности выполнения задач, запоминаемости интерфейса и отсутствия ошибок при использовании.

Тест-кейс тестирования удобства пользования представлен в таблице №5.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				31

Тест-кейс тестирования на удобство пользования

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на интуитивность интерфейса	Попросить новых пользователей распознать жесты с помощью приложения и переключать язык без предварительного инструктажа.	Пользователи могут интуитивно понять интерфейс и выполнить задачи без внешней помощи.	Пользователи могут интуитивно понять интерфейс и выполнить задачи без внешней помощи.
Тест на запоминаемость интерфейса	Попросить пользователей распознать жесты с помощью программы и переключать язык после перерыва в использовании программы.	Пользователи могут легко вспомнить, как использовать программу после перерыва.	Пользователи могут легко вспомнить, как использовать программу после перерыва.
Тест на доступность для пользователей с ограниченными возможностями	Проверить, насколько легко программой пользоваться людям с нарушениями слуха.	Программа доступна и удобна для использования пользователями с нарушениями слуха.	Программа доступна и удобна для использования пользователями с нарушениями слуха.

3.4.6 Нагрузочное тестирование

Нагрузочное тестирование — это процесс оценки способности программного обеспечения справляться с предполагаемым или максимальным объёмом работы. Нагрузочное тестирование может включать проверку способности программы обрабатывать высокий поток входящих видеоданных, одновременные запросы на распознавание жестов и способность системы сохранять стабильность и производительность под нагрузкой.

Тест-кейс нагрузочного тестирования представлен в таблице № 6.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	32

Тест-кейс для нагрузочного тестирования

Описание	Порядок действий	Ожидаемый результат	Фактический результат
Тест на максимальное количество одновременных распознаваний жестов	Подать на вход программы 3 видеопотока с жестами одновременно.	Программа корректно обрабатывает все потоки, не теряя в производительности и не выдавая ошибок.	Программа корректно обрабатывает все потоки, не теряя в производительности и не выдавая ошибок.
Тест на производительность при ограничении ЦП и ОЗУ	Ограничить доступные системные ресурсы ЦП до 2 ядер и ОЗУ до 4 ГБ и запустить программу.	Программа эффективно распределяет доступные ресурсы, сохраняя работоспособность при их ограничении.	Программа эффективно распределяет доступные ресурсы, сохраняя работоспособность при их ограничении.
Тест на длительную нагрузку	Запустить программу на 1 час с постоянной обработкой видеопотоков.	Программа стабильно работает без утечек памяти и снижения производительности в течение всего теста.	Программа стабильно работает без утечек памяти и снижения производительности в течение всего теста.

Результаты всех видов тестирования показали, что разрабатываемое приложение успешно справляется с поставленными задачами и соответствует заданным требованиям. Это свидетельствует о надежности, безопасности и высокой производительности приложения, а также о его удобстве и доступности для широкого круга пользователей.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	33

ЗАКЛЮЧЕНИЕ

В ходе работы над дипломным проектом «Разработка приложения для конвертации видеопотока в бегущую строку для распознавания языка жестов» были успешно выполнены следующие задачи:

- разработаны алгоритмы для распознавания языка жестов с использованием технологий компьютерного зрения и машинного обучения;
- создан пользовательский интерфейс, обеспечивающий удобное и интуитивно понятное взаимодействие с программой;
- реализована функция перевода распознанных жестов в текст и их дальнейшее преобразование в аудио формат;
- проведено тестирование программы для проверки её функциональности и производительности.

Проект достиг цели создания эффективного и доступного средства для перевода языка жестов, что значительно улучшает возможности общения для людей с нарушениями слуха.

В результате проделанной работы была разработана программа «Переводчик языка жестов», которая демонстрирует высокую точность распознавания жестов и их перевода в текст и аудио- форматы. Программа может быть использована в различных сферах, где необходимо облегчить общение между людьми с нарушениями слуха и остальным обществом.

В перспективе проект может быть дополнен новыми функциями, такими как распознавание жестов в условиях сложной освещенности, улучшение точности перевода и расширение словарного запаса.

Цели проекта были достигнуты, и разработанный «Переводчик языка жестов» вносит значительный вклад в социальную инклюзию и обеспечение равных возможностей для всех членов общества.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				34

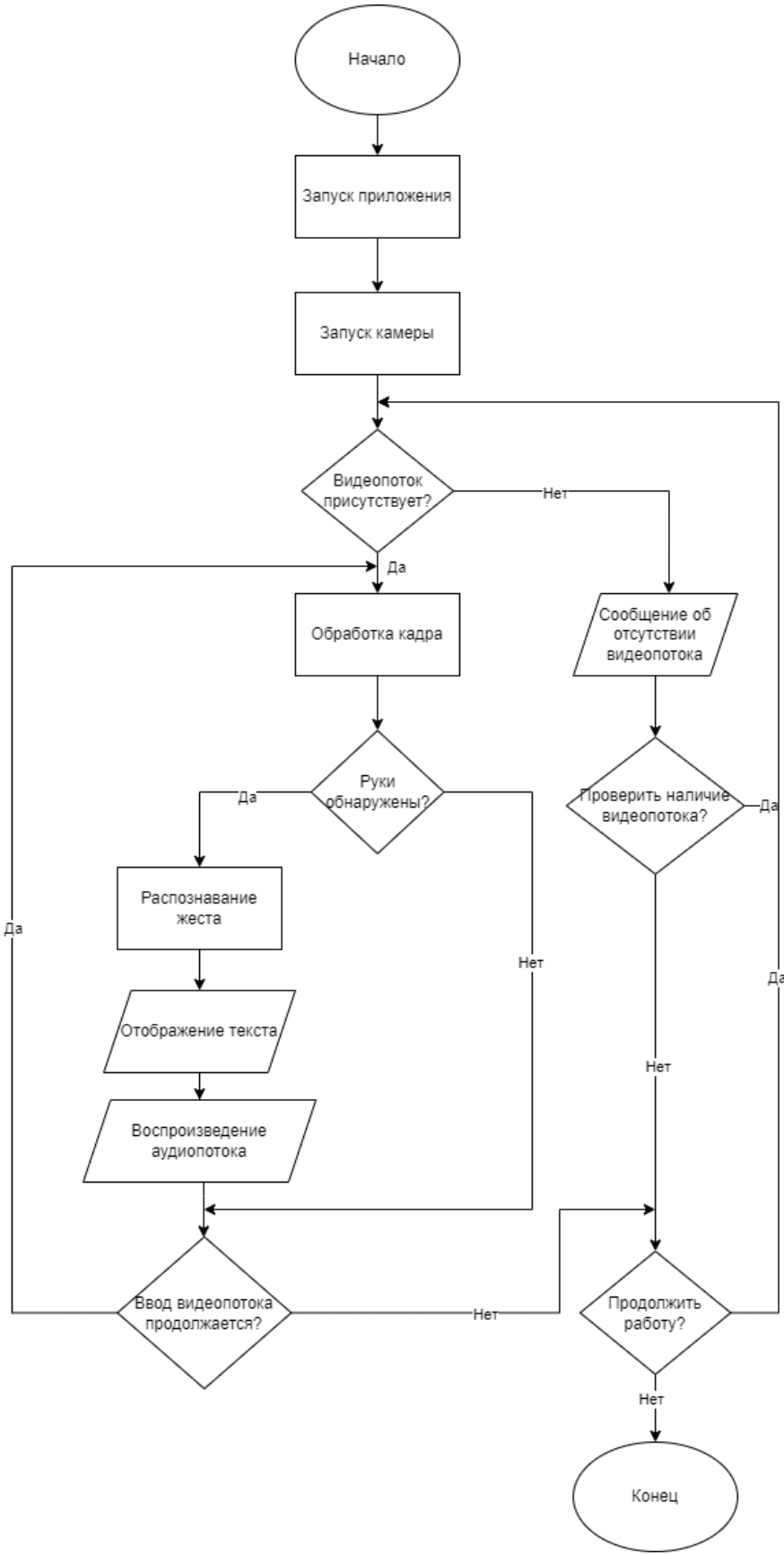
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://unesdoc.unesco.org/ark:/48223/pf0000190431_rus – ИКТ для инклюзивного образования.
2. <https://nauchkor.ru/pubs/assistivnye-tehnologii-v-sotsialnoy-inklyuzii-detey-s-narusheniyami-sluha-605ee01accefd000190ab15> – ассистивные технологии в социальной инклюзии детей с нарушениями слуха.
3. <https://cyberleninka.ru/article/n/issledovanie-yazyka-zhesta-dlya-tsifrovogo-perevoda-obzornyy-analiz/viewer> – исследование языка жеста для цифрового перевода: обзорный анализ.
4. <https://slait.ai/> – официальный сайт приложения «SLAIT».
5. <http://mimix.me/> – официальный сайт приложения «Mimix Speech To Signs».
6. <https://edwize.org/ace-asl/> – официальный сайт приложения «Ace ASL».
7. <https://wireframe.cc/> – сайт для проектирование интерфейсов «Wireframe.cc».
8. Мартелли Алекс, Рейвенскрофт Анна, Холден Стив Python. Справочник. Полное описание языка. - 3-е изд. - Диалектика, 2019. - 896 с.
9. Джозеф Хауз OpenCV Computer Vision with Python. - Бирмингем: Packt Publishing, 2024. - 112 с.
10. https://scikit-learn.ru/getting_started/ – документация по библиотеке «scikit-learn».
11. <https://docs.python.org/3.10/library/pickle.html> – документация по библиотеке « pickle ».
12. <https://developers.google.com/mediapipe/> – документация по библиотеке «MediaPipe».
13. <https://pillow.readthedocs.io/en/stable/> – документация по библиотеке «Pillow».
14. Алан Д. Мур Python GUI Programming with Tkinter. - 2-е изд. - Бирмингем: Packt Publishing, 2021. - 664 с.

Изм.	Лист	№ докум	Подпись	Дата		Лист
	Разраб.	Сова С.С.			ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	35

15. <https://pygame-docs.website.yandexcloud.net/> – документация по библиотеке «Pygame».
16. Базоев В. З., Егорова И. А., Гаврилова Г. Н. Словарь русского жестового языка. - 6-е изд. - М.: Флинта, 2023. - 528 с.
17. Тара Адамс American Sign Language Dictionary. - Беркли: Rockridge Press, 2022. - 334 с.

Изм.	Лист	№ докум	Подпись	Дата	ДП-09.02.07.ЩПКО-02-20.18.06.24.ПЗ	Лист
Разраб.		Сова С.С.				36



Исходный код обработки видеопотока и распознавания жестов

```
def update_frame():
    global last_character, text_output
    ret, frame = cap.read()
    if not ret:
        window.after(10, update_frame)
        return
    H, W, _ = frame.shape
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(frame_rgb)
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            data_aux, x_, y_ = [], [], []
            for i, landmark in enumerate(hand_landmarks.landmark):
                x, y = landmark.x, landmark.y
                x_.append(x)
                y_.append(y)
            data_aux.extend([x - min(x_), y - min(y_)])
            prediction = current_model.predict([np.asarray(data_aux)])
            predicted_character = labels_dict[int(prediction[0])]
            pil_image = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
            frame = cv2.cvtColor(np.array(pil_image), cv2.COLOR_RGB2BGR)
            if predicted_character != last_character:
                play_sound(predicted_character)
                last_character = predicted_character
            output_label.config(text=text_output)
            update_class_history(predicted_character)
            cv_img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            img = Image.fromarray(cv_img)
            imgtk = ImageTk.PhotoImage(image=img)
            label.imgtk = imgtk
            label.configure(image=imgtk)
            window.after(10, update_fr
```