

УДК 004.93

АРХИТЕКТУРНАЯ МОДЕРНИЗАЦИЯ ПРИЛОЖЕНИЯ GESTOTALK

А. С. МИЩЕНКО, И. И. МИЩЕНКО

Белорусско-Российский университет

Могилев, Беларусь

Первая версия Gestotalk распознавала статические жесты – положения рук, которые не меняются во времени. Для сравнения использовались координаты 21 точки кисти, полученные с помощью библиотеки MediaPipe. Однако программа имела ряд проблем. Во-первых, камера и распознавание работали в одном потоке с интерфейсом, из-за чего окно «замирало» на время обработки каждого кадра. Во-вторых, распознанный символ записывался в текстовый файл на каждом кадре, что вызывало множественные повторы и тормозило систему. В-третьих, жесты, требующие движения руки, – «й», «щ», «ё» – вообще не распознавались, т. к. система умела сравнивать только неподвижные жесты. Наконец, озвучка текста работала только через онлайн-сервис Google, требовалось постоянное подключение к интернету и сам процесс озвучки блокировал интерфейс программы [1].

Для решения первой проблемы в версии Gestotalk 2.0 захват видео и работа детектора рук MediaPipe перенесены в отдельный фоновый поток – CameraThread, реализованный на базе класса QThread из библиотеки PyQt5. Детектор создаётся один раз при запуске, а не на каждом кадре, как раньше. Готовые кадры с найденными точками руки передаются в основное окно через механизм «сигналов» – это способ безопасной передачи данных между потоками. В результате интерфейс перестаёт зависать и частота кадров держится на уровне 30 FPS [2].

Чтобы система не зависела от того, где находится рука в кадре и на каком расстоянии от камеры, координаты точек нужно «нормализовать» – привести к единому масштабу. В первой версии это делалось в обычном цикле Python с поиском минимума и максимума, что работало медленно. В версии 2.0 используется библиотека NumPy, которая умеет обрабатывать массивы чисел за одну операцию. Система сдвигает все точки так, чтобы запястье оказалось в начале координат, а затем делит расстояния на размер ладони – расстояние от запястья до основания среднего пальца. Так рука описывается относительными величинами и распознавание не портится, если пользователь подвинул руку ближе к камере или сдвинул её в сторону.

В Gestotalk 1.0 жест регистрировался на каждом кадре, где рука была видна. Если пользователь показывал букву «А» в течение секунды, в текст попадало 30 символов «А». В версии 2.0 введён GestureBuffer – промежуточный накопитель, который собирает результаты с нескольких кадров подряд и фиксирует жест только тогда, когда он повторяется достаточное число раз (например, 3 раза из 5). Это устраняет ложные срабатывания, когда детектор на один кадр ошибочно видит другую букву. Для служебных команд («ввод»,

«пробел», «удалить») добавлен Cooldown – принудительная пауза после выполнения команды, чтобы она не сработала несколько раз подряд.

Буквы «й», «щ», «ё» в дактильном алфавите нельзя показать неподвижной рукой: они требуют движения руки. Чтобы научить программу распознавать такие жесты, обычно нужно собирать тысячи примеров и обучать сложную нейросеть. В версии 2.0 использован другой подход – алгоритм DTW (Dynamic Time Warping, динамическая трансформация времени). Этот алгоритм сравнивает две траектории движения, даже если они выполнены с разной скоростью. При распознавании система сравнивает текущее движение с сохранённым эталоном. Если кривые похожи – жест распознан. Это позволило добавить динамические жесты без сбора большого набора данных и без обучения нейросети.

В первой версии программы текст хранился в файле, который открывался и перезаписывался на каждом кадре. Это вызывало задержки из-за постоянного обращения к жёсткому диску. В версии 2.0 введён TextEngine – внутренний блок программы, который хранит набранный текст в оперативной памяти компьютера. Он работает быстро и позволяет добавлять, удалять и очищать символы без обращения к диску.

Синтез речи (превращение текста в голос) также изменён. Раньше использовался только онлайн-сервис gTTS (Google Text-to-Speech), который требовал интернет и останавливал работу интерфейса на время загрузки. Теперь приоритет отдан офлайн-движку pyttsx3 – он работает полностью на компьютере пользователя, без сети. Озвучка выполняется в отдельном фоновом потоке TTSThread, поэтому видео и интерфейс не прерываются.

Рефакторинг Gestotalk 2.0 кардинально повысил производительность и надёжность приложения. Многопоточность и векторная нормализация устранили зависания интерфейса, алгоритм DTW и дедупликация жестов расширили словарь и повысили точность, а перенос буфера в оперативную память и офлайн-синтез речи обеспечили автономную работу без интернета. Архитектура приложения стала модульной, что открывает перспективы для дальнейшего масштабирования функционала.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. **Роголев, Д. В.** Применение технологий компьютерного зрения и машинного обучения для распознавания и перевода жестового языка / Д. В. Роголев, А. С. Голяс, И. И. Мищенко // Вестник Белорусско-Российского университета. – 2026. – № 1 (90). – С. 105–116.
2. **Голяс, А. С.** Система распознавания жестового языка на основе анализа видеопотока в реальном времени / А. С. Голяс // Новые материалы, оборудование и технологии в промышленности : материалы Междунар. науч.-техн. конф. – Могилев : Бел.-Рос. ун-т, 2025. – С. 152.