ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМА ПОИСКА ОБЪЕКТОВ НА ВИДЕО С ПОМОЩЬЮ ГОЛОСОВЫХ КОМАНД

Мальченко А.Е., Шапутько А.Е., студенты

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Перцев Д.Ю. – канд. техн. наук, доцент

В работе рассмотрена архитектура системы детекции объектов в видеопотоке с возможностью голосового выбора целевых классов для обнаружения. Проведён анализ интеграции нейросетевой модели YOLO-World и системы распознавания речи Vosk, реализована многопоточная обработка для обеспечения работы в реальном времени. Представлены результаты тестирования производительности и точности системы.

Детекция объектов в контексте обработки видеопотока – это процесс автоматического выделения и классификации объектов на кадрах видео в реальном времени. Целью работы ставится создание интуитивного интерфейса взаимодействия человека с системой компьютерного зрения, позволяющего динамически менять целевые классы для обнаружения без перезагрузки модели [1].

Для детекции объектов разработан ряд нейросетевых архитектур. Из-за того, что универсального решения для всех сценариев не существует, при разработке системы необходимо проведение анализа методов и понимание специфики выбранной области, на основе чего выбирается архитектура.

Целью данной работы является разработка системы детекции объектов с голосовым управлением на русском языке, проведение сравнительных тестов производительности на реальных данных и последующая оптимизация для работы в реальном времени.

Алгоритм детекции на базе YOLO-World. YOLO-World (You Only Look Once – World) – это архитектура нейросети для детекции объектов с открытым словарём, позволяющая детектировать объекты по текстовому описанию без дообучения модели [2]. В отличие от классических YOLO-моделей, обученных на фиксированном наборе классов, YOLO-World использует механизм сопоставления текстовых эмбеддингов с визуальными признаками, что позволяет динамически менять список детектируемых объектов.

Данный подход позволяет поделить процесс детекции на два этапа:

1. Извлечение признаков: свёрточная сеть извлекает визуальные признаки из входного кадра.
2. Сопоставление с текстом: механизм внимания сопоставляет визуальные признаки с эмбеддингами целевых классов.

Такой подход корректно работает при наличии небольшого количества целевых классов, что подразумевает фокусировку на конкретных объектах. Большое количество классов приводит к снижению точности и производительности. Для решения данной проблемы существуют следующие способы:

1. Применение фильтрации по порогу уверенности для отсечения ложных срабатываний.
2. Использование YOLO-World для ограничения списка детектируемых объектов.

При тестировании метод применялся для решения задачи поиска конкретных объектов по голосовой команде.

Алгоритм распознавания речи на базе Vosk. Vosk – это оффлайн-система распознавания речи с открытым исходным кодом, основанная на архитектуре Kaldi [3]. Данный метод основан на построении статистической модели языка и акустической модели для преобразования аудиосигнала в текст. Аудиосигналы, имеющие близкие спектральные характеристики, объединяются в фонемы, которые затем сопоставляются со словами словаря.

Для упрощения описания процесса распознавания была введена модель декодирования, где максимумы функции вероятности расположены в точках соответствия аудиосегментов слов. Аудиосегменты, которые принадлежат одному слову, объединяются в текстовую строку.

Дополнительно при работе с данным алгоритмом необходимо выбрать языковую модель под конкретную задачу. Если в качестве модели выбирается русскоязычная (vosk-model-small-ru-0.22), система будет корректно распознавать команды на русском языке. Соответственно, при выборе другой модели, точность распознавания на идентичном аудиопотоке изменится.

Недостатком выбора такой модели является то, что алгоритм требует предварительной настройки порога чувствительности для активации распознавания. Преимущество Vosk заключается в том, что он работает оффлайн и обеспечивает задержку менее 200 мс.

Многопоточная архитектура системы. Развертывания всей системы подразумевает разделение ее на независимые потоки выполнения. Для обеспечения работы в реальном времени выбран подход с четырьмя потоками:

1. GUI-поток обработка событий интерфейса PyQt6 [4], отрисовка кадров.
2. Видеопоток захват кадров, детекция YOLO.
3. Аудио-поток: захват аудио, распознавание Vosk, обработка команд.

4. Поток передачи данных по UART на платформу.

В результате формируется система, где каждый компонент выполняет свою задачу без блокировки остальных. Данный подход полезен для минимизации задержек в конвейере обработки. Одним из способов синхронизации потоков является использование `threading.Lock` для доступа к общим данным и `pyqtSignal` для безопасной передачи данных в интерфейс.

При решении задач в контексте обработки видеопотока данный подход позволяет достичь стабильной частоты кадров. Для корректной работы требуется передача кадров между потоками через копирование данных (QPixmap.copy()), что предотвращает гонки данных.

Сравнительный анализ результатов работы методов проводился на аппаратных системах, представленных в таблице 1.

Таблица 1 – Аппаратные системы

Параметр	**Ноутбук 1**	**Ноутбук 2**
Процессор	Intel Core i5	AMD Ryzen 5 4600H
ОЗУ	8 ГБ	16 ГБ
Видеокарта	Интегрированная	NVIDIA GeForce GTX 1650 Ti (4 ГБ)
Распознавание речи, с.	0.14	0.14
Полный цикл: аудио и детекция	4.20	3.92

Результаты проведенного анализа представлены в таблице 2.

Таблица 2 – Результаты анализа

Метрика	**Ноутбук 1**	**Ноутбук 2**
Общий FPS системы (640×480), кадров/с.	4–20	24.27
Время обработки кадра, мс.	30–40	25–35
Обработка аудио-блока, мс.	46.8–74.0	40–65
Распознавание речи, с.	0.14	0.14
Полный цикл: аудио и детекция	4.20	3.92

Метод оптимизации модели удовлетворительным образом решает задачу повышения производительности. Алгоритм перехода на nano-версию модели детекции объектов и распознавания голоса относительно точно сохраняет точность детекции при значительном приросте FPS (было 10 FPS, стало 20-24 FPS).

Алгоритмы YOLO-World и Vosk имеют схожий результат при интеграции с общими слабыми местами: оба метода склонны к снижению точности при высоком уровне шума во входных данных. При работе с обоими алгоритмами необходима качественная предобработка, если техническими требованиями определена высокая точность распознавания.

Метод динамической смены классов при определённых условиях может создавать избыточную нагрузку: частые вызовы `set_classes()` приводят к перезагрузке весов модели, что малозаметно при редких командах, но становится критичным при частом переключении.

Заключение. Таким образом, решение о применении каждого алгоритма необходимо принимать на основе специфики задачи и условий эксплуатации системы. Обработка видеопотока в реальном времени – это процесс с большим количеством нестабильных условий, под которые постоянно необходимо подстраиваться и минимизировать их последствия. Поэтому оптимальным выбором при решении задач по детекции объектов с голосовым управлением является комбинирование всех алгоритмов под определенную задачу.

Список использованных источников:

1. What is object detection [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.ibm.com/think/topics/object-detection – Дата доступа: 25.03.2026.

2. YOLO-World: Real-Time Open-Vocabulary Object Detection [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/AILab-CVC/YOLO-World – Дата доступа: 25.03.2026.

3. Vosk: Offline speech recognition toolkit [Электронный ресурс]. – Электронные данные. – Режим доступа: https://alphacephei.com/vosk/ – Дата доступа: 25.03.2026.

4. PyQt6 Documentation [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.riverbankcomputing.com/static/Docs/PyQt6/ – Дата доступа: 25.03.2026.