**ОТЧЕТ О ТЕСТИРОВАНИИ.**

**Система обнаружения людей на изображениях с использованием YOLOv4**

Выполнил начинающий тестировщик

Дашкин С.М.

**Оглавление**

[**1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ** 3](#_Toc204201628)

[**2. СОСТАВ КОМАНДЫ ТЕСТИРОВАНИЯ** 4](#_Toc204201629)

[**3. МЕТОДОЛОГИЯ И ПРОЦЕССЫ ТЕСТИРОВАНИЯ** 4](#_Toc204201630)

[**4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ** 5](#_Toc204201631)

[**5. ОБНАРУЖЕННЫЕ ДЕФЕКТЫ** 6](#_Toc204201632)

[**6. АНАЛИЗ ПОКРЫТИЯ ТЕСТИРОВАНИЯ** 7](#_Toc204201633)

[**7. ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ** 8](#_Toc204201634)

[**8. РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ** 9](#_Toc204201635)

[**9. ОЦЕНКА ГОТОВНОСТИ К РЕЛИЗУ** 10](#_Toc204201636)

[**10. МЕТРИКИ И СТАТИСТИКА** 11](#_Toc204201637)

[**11. ПЛАНЫ НА СЛЕДУЮЩУЮ ИТЕРАЦИЮ** 12](#_Toc204201638)

[**12. ЗАКЛЮЧЕНИЕ** 13](#_Toc204201639)

[**СПИСОК ЛИТЕРАТУРЫ** 14](#_Toc204201640)

Система обнаружения людей на изображениях с использованием YOLOv4

Версия отчета: 1.0  
Дата составления: 23.07.2025  
Период тестирования: 15.07.2025 - 23.07.2025  
Тип отчета: Финальный отчет о тестировании

# **1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ**

1.1 Описание системы

Тестируемая система представляет собой Windows Forms приложение для обнаружения людей на статических изображениях с использованием предварительно обученной нейронной сети YOLOv4. Система включает в себя:

* Графический интерфейс пользователя (Form1.cs)
* Модуль детекции людей (EmguHumanDetector.cs)
* Алгоритмы обработки изображений с использованием библиотеки EmguCV

1.2 Цель тестирования

Проверить функциональность, производительность и надежность системы обнаружения людей на изображениях, оценить готовность к промышленному использованию.

# **2. СОСТАВ КОМАНДЫ ТЕСТИРОВАНИЯ**

| Роль | ФИО | Ответственность |
| --- | --- | --- |
| Test Manager | Дашкин С.М. | Планирование и координация тестирования |
| QA Engineer | Дашкин С.М. | Выполнение функционального тестирования |
| Performance Tester | Дашкин С.М. | Тестирование производительности |

# **3. МЕТОДОЛОГИЯ И ПРОЦЕССЫ ТЕСТИРОВАНИЯ**

3.1 Применяемые методы тестирования

* Функциональное тестирование: Проверка основных функций загрузки, обработки и сохранения изображений
* Тестирование интерфейса: Проверка корректности работы GUI элементов
* Тестирование совместимости: Проверка работы с различными форматами изображений
* Негативное тестирование: Проверка поведения системы при некорректных входных данных

3.2 Инструменты тестирования

* Среда разработки: Visual Studio 2022
* Фреймворк: .NET 9.0
* Библиотеки: EmguCV для обработки изображений
* Модель ИИ: YOLOv4 (yolov4.weights, yolov4.cfg)

# **4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

4.1 Функциональное тестирование

4.1.1 Тестирование загрузки изображений

Статус: Passed

* Система корректно загружает изображения через OpenFileDialog
* Поддерживаются стандартные форматы (PNG, JPG, BMP)
* Обработка ошибок при загрузке некорректных файлов реализована

4.1.2 Тестирование инициализации модели

Статус: частично Passed

* Автоматический поиск файлов модели в 7 различных директориях
* Ручной выбор файлов модели при их отсутствии
* Проблема: Зависимость от наличия файлов yolov4.weights и yolov4.cfg

4.1.3 Тестирование детекции объектов

Статус: Passed

* Обнаружение людей на изображениях работает корректно
* Настройка порога уверенности (1-50%)
* Режим высокой чувствительности для сложных случаев
* Визуализация результатов с цветовой индикацией уверенности

4.1.4 Тестирование сохранения результатов

Статус: Passed

* Сохранение обработанных изображений в формате PNG
* Автоматическое добавление суффикса "\_processed" к имени файла

4.2 Статистика выполнения тест-кейсов

Модуль загрузки изображений: 15/15 (100%)

Модуль детекции: 12/15 (80%)

Модуль сохранения: 8/8 (100%)

Пользовательский интерфейс: 20/22 (91%)

ИТОГО: 55/60 (92%)

# **5. ОБНАРУЖЕННЫЕ ДЕФЕКТЫ**

5.1 Критические дефекты

Количество: 0

5.2 Высокоприоритетные дефекты

Количество: 2

DEF-001: Зависимость от файлов модели

* Приоритет: Высокий
* Описание: Приложение не может функционировать без файлов yolov4.weights (>200MB) и yolov4.cfg
* Влияние: Блокирует запуск приложения при отсутствии файлов модели
* Статус: Открыт
* Рекомендация: Реализовать автоматическую загрузку модели или включить легковесную альтернативу

DEF-002: Отсутствие валидации входных изображений

* Приоритет: Высокий
* Описание: Недостаточная проверка размера и формата загружаемых изображений
* Влияние: Возможные ошибки при обработке нестандартных файлов
* Статус: Открыт

5.3 Среднеприоритетные дефекты

Количество: 3

DEF-003: Производительность при больших изображениях

* Приоритет: Средний
* Описание: Заметное снижение производительности при обработке изображений > 2MB
* Статус: Открыт

5.4 Низкоприоритетные дефекты

Количество: 1

# **6. АНАЛИЗ ПОКРЫТИЯ ТЕСТИРОВАНИЯ**

6.1 Функциональное покрытие

* Покрытие кода: 75-85%
* Покрытие требований: 90-95%
* Покрытие интерфейсов: 100%

6.2 Непокрытые области

1. Тестирование производительности: Отсутствуют нагрузочные тесты
2. Тестирование безопасности: Не проводилось
3. Тестирование совместимости: Ограниченное тестирование ОС
4. Юзабилити тестирование: Не проводилось

# **7. ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ**

7.1 Время обработки изображений

*Конкретные метрики производительности отсутствуют, тк нагрузочные тесты не проводились.*

Требуемые метрики:

* Время инициализации модели: ~X секунд
* Время обработки изображения 1920x1080: ~X секунд
* Время обработки в режиме высокой чувствительности: +X% к базовому времени
* Использование памяти: ~X MB
* Загрузка CPU: ~X%

Рекомендуемые бенчмарки:

|  |  |  |
| --- | --- | --- |
| Размер изображения | Время обработки | Режим ВЧ |
| 800x600 | <2 сек | <4 сек |
| 1920x1080 | <5 сек | <10 сек |
| 4K (3840x2160) | <15 сек | <30 сек |

# **8. РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ**

8.1 Результаты регрессионных тестов

*Регрессионное тестирование не проводилось.*

Требуемая информация:

* Количество регрессионных тест-кейсов
* Результаты выполнения после изменений кода
* Сравнение с предыдущими версиями
* Выявленные регрессии

# **9. ОЦЕНКА ГОТОВНОСТИ К РЕЛИЗУ**

9.1 Критерии готовности

| Критерий | Статус | Комментарий |
| --- | --- | --- |
| Функциональность | Работает | Основные функции работают |
| Стабильность | Работает, но с условием | Есть зависимость от внешних файлов |
| Производительность | Тестирование не проводилось | Требует дополнительного тестирования |
| Документация | Не написана | Отсутствует пользовательская документация |
| Критические дефекты | Не выявлены | Отсутствуют |

9.2 Рекомендации

РЕКОМЕНДАЦИЯ ПО РЕЛИЗУ: УСЛОВНО ГОТОВ

Условия для релиза:

1. Решить проблему с зависимостью от файлов модели (DEF-001)
2. Добавить валидацию входных данных (DEF-002)
3. Создать пользовательскую документацию
4. Провести дополнительное тестирование производительности

Риски при релизе без доработок:

* Проблемы с развертыванием из-за отсутствия файлов модели
* Возможные сбои при обработке нестандартных изображений
* Неудовлетворительная производительность на слабых машинах

# **10. МЕТРИКИ И СТАТИСТИКА**

10.1 Общая статистика

*Точные метрики не зафиксированы.*

Требуемые метрики:

Общее время тестирования: X человеко-часов

Количество тест-кейсов: X

Плотность дефектов: X дефектов на 1000 строк кода

Эффективность обнаружения дефектов: X%

Скорость исправления дефектов: X дней в среднем

10.2 Диаграммы и графики

*Что можно было бы дописать:*

* Динамика обнаружения и исправления дефектов
* Распределение дефектов по приоритетам
* Прогресс выполнения тест-кейсов
* Тренд готовности к релизу

# **11. ПЛАНЫ НА СЛЕДУЮЩУЮ ИТЕРАЦИЮ**

11.1 Приоритетные задачи

1. Исправление критических дефектов
   * Решение проблемы с файлами модели
   * Улучшение валидации входных данных
2. Дополнительное тестирование
   * Нагрузочное тестирование
   * Тестирование на различных конфигурациях
   * Юзабилити тестирование
3. Документирование
   * Создание пользовательского руководства
   * Техническая документация для разработчиков

11.2 Долгосрочные улучшения

* Реализация поддержки видеофайлов
* Добавление пакетной обработки изображений
* Интеграция с облачными сервисами
* Улучшение точности детекции

# **12. ЗАКЛЮЧЕНИЕ**

Система обнаружения людей продемонстрировала хорошую функциональность в основных сценариях использования. Архитектура кода структурирована и понятна, использованы современные технологии и библиотеки.

Сильные стороны:

* Стабильная работа основного функционала
* Гибкие настройки чувствительности детекции
* Интуитивно понятный интерфейс
* Качественная визуализация результатов

Области для улучшения:

* Зависимость от внешних файлов модели
* Недостаточное тестирование производительности
* Отсутствие комплексной документации
* Ограниченная обработка ошибок

Итоговая оценка: Система готова к ограниченному использованию после устранения выявленных высокоприоритетных дефектов и создания необходимой документации.

**Отчет подготовлен:** QA Team Lead, Дашкин С.М.  
**Дата:** 23.07.2025  
**Версия отчета:** 1.0

# **СПИСОК ЛИТЕРАТУРЫ**

* 1. Ефремова Д. [Создание понятных отчетов о тестировании](https://habr.com/ru/companies/performance_lab/articles/207512/).
  2. QA\_Bible [Виды отчётов(Reports)](https://vladislaveremeev.gitbook.io/qa_bible/testovaya-dokumentaciya-i-artefakty-test-deliverablestest-artifacts/vidy-otchetov-reports).
  3. Skypro [Отчет о тестировании сайта: пример и советы](https://sky.pro/wiki/profession/otchet-o-testirovanii-sajta-primer-i-sovety/)