**Проверка деталей на брак**

1. Введение

Данное техническое задание разработано для создания системы автоматической проверки деталей на наличие дефектов и брака. Система будет применять методы визуального контроля, что повысит точность в идентификации дефектов и позволит уменьшить человеческий фактор в процессе контроля качества. В условиях современного производства, где высокие требования к качеству продукции становятся стандартом, внедрение автоматизированной системы контроля является необходимым шагом для повышения конкурентоспособности предприятия.

2. Объект проверки

- Механические детали: детали, используемые в машинах и механизмах (например, шестерни, болты, гайки).

- Электронные компоненты: платы, микроконтроллеры, соединители, резисторы и конденсаторы.

- Сборки узлов: узлы, которые состоят из нескольких деталей (например, сборки двигателей, трансмиссий).

3. Требования к системе

**Функциональные требования:**

- Автоматическая идентификация дефектов: система должна быть способна классифицировать дефекты (царапины, трещины, геометрические отклонения и т.д.).

- Визуальный осмотр: использование камер с высоким разрешением и возможностью изменения угла обзора для детального осмотра.

- Интеграция с ERP: необходимо обеспечить взаимодействие с существующей системой учета и управления, чтобы данные о дефектах автоматически перемещались в систему.

- Отчетность: автоматическое создание отчетов, которые содержат информацию о проверенных партиях, количестве выявленных дефектов и их типах.

**Нефункциональные требования:**

- Надежность системы: минимальная вероятность ложных срабатываний (не более 2%).

- Скорость обработки данных: не менее 30 деталей в минуту.

- Интерфейс пользователя: графический интерфейс, который будет предельно простым и понятным для операторов. Возможность настройки параметров проверки.

4. Процесс проверки

**1. Подготовка деталей:**

- Определение времени и места для проведения проверки.

- Подготовка деталей к съемке (например, очистка от загрязнений).

**2. Захват изображения:**

- Установка камеры на производственной линии.

- Настройка освещения для снижения бликов и улучшения качества изображения.

**3. Обработка изображений:**

- Использование алгоритмов компьютерного зрения для анализа изображений.

- Обучение нейронных сетей на примерах дефектных и недефектных деталей для повышения точности распознавания.

**4. Сравнение и анализ:**

- Сравнение полученных данных с установленными нормами и стандартами качества.

- Учет исторических данных для улучшения алгоритмов.

**5. Выгрузка результатов:**

- Автоматизация процесса записи результатов проверки в базу данных.

- Формирование отчетов и уведомлений для ответственных лиц.

5. Описание этапов реализации

**1. Анализ текущих процессов:** изучение существующих методов проверки и определение узких мест.

**2. Выбор оборудования:** исследование рынка на предмет камер, компьютеров и ПО, которое будет использоваться.

**3. Разработка алгоритмов обработки изображений:** создание и тестирование алгоритмов на исторических данных.

**4. Тестирование системы:** запуск пробных партий и коррекция алгоритмов.

**5. Обучение персонала:** работа с сотрудниками для достижения понимания работы системы и её управления.

**6. Внедрение системы:** интеграция в производственный процесс и финальное тестирование.

6. График выполнения работ

- Месяцы: 1-3 — анализ и проектирование; 4-5 — разработка и тестирование; 6 — внедрение и обучение.

7. Бюджет

- Оборудование: 300,000 рублей (камеры, компьютеры)

- Программное обеспечение: 100,000 рублей (лицензионные версии ПО)

- Обучение персонала: 50,000 рублей (Аренда зала, материалы)

- Прочие расходы: 50,000 рублей (транспорт, сопутствующие материалы)

8. Заключение

Успешная реализация данного проекта позволит значительно повысить качество продукции, уменьшить количество бракованных деталей и оптимизировать производственные процессы.

Внедрение системы визуального контроля создаст условия для более эффективного управления качеством, что, в свою очередь, положительно скажется на удовлетворенности клиентов и конкурентоспособности компании. Благодаря автоматизации процессов контроля, производство сможет адаптироваться к изменениям на рынке быстрее и с меньшими затратами, что обеспечит устойчивый рост и развитие предприятия.

**Язык и технологии для автоматической проверки деталей**

**Python**

**Почему:** Python — это высокоуровневый язык программирования, известный своей простотой и читаемостью. Это делает его идеальным выбором для быстрого прототипирования и разработки решений автоматической проверки деталей.

**Преимущества использования Python:**

- Простота и удобочитаемость: Синтаксис Python интуитивно понятен, что позволяет разработчикам быстро ознакомиться с кодом и легко его поддерживать.

- Широкий спектр библиотек: Python имеет обширное сообщество и множество библиотек для различных задач:

- Обработка данных: Библиотеки, такие как Pandas и NumPy, позволяют легко работать с большими наборами данных, проводить их анализ и манипуляции.

- Машинное обучение: Библиотеки, такие как Scikit-learn, TensorFlow и Keras, предоставляют инструменты для построения моделей машинного обучения, что актуально для анализа дефектов и предсказания их возникновения.

- Обработка изображений: OpenCV и PIL позволяют выполнять различные операции над изображениями, что важно в контексте визуальной проверки деталей.

- Сообщество и поддержка: Сильное сообщество разработчиков и множество ресурсов делают обучение Python и решение возникающих проблем быстрее и проще.

- Кроссплатформенность: Python работает на различных операционных системах (Windows, Mac, Linux), что облегчает развертывание решений в разных средах.

- Интеграция с другими языками: Python можно легко интегрировать с C/C++, Java и другими языками, что позволяет использовать его в составе более комплексных систем.

**Примеры применения Python в автоматической проверке деталей:**

- Анализ дефектов: Создание моделей машинного обучения для обнаружения отклонений в характеристиках деталей на основе исторических данных и результатов проверок.

- Автоматизация процессов: Написание скриптов для автоматизации сбора и обработки данных о проверках, что ускоряет процесс принятия решений.

- Визуализация данных: Использование библиотек, таких как Matplotlib и Seaborn, для визуализации результатов проверок, что помогает быстрее идентифицировать проблемы и тенденции.

