



LEGO Package Verifier - Documento de Diseño

Versión 1.0

Febrero 2020

LEGO Package Verifier

Documento de Diseño

Nicolás Camacho Plazas, Mateo Florido Sanchez

13 de febrero de 2020

Documento de Diseño

Versión v1.0, Febrero 2020

Authors - Nicolás Camacho Plazas, Mateo Florido

This entire document is licensed under GNU General Public License v3.0. Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

LEGO System A/S, DK-7190 Billund, Denmark. LEGO, the LEGO logo, the Minifigure, DUPLO, LEGENDS OF CHIMA, NINJAGO, BIONICLE, MINDSTORMS and MIXELS are trademarks and copyrights of the LEGO Group. ©2019 The LEGO Group. All rights reserved.

Índice general

| | | |
|----------|---|----------|
| 1 | Definición del Problema | 4 |
| 1.1 | Solución a Proponer | 4 |
| 1.2 | Alcance | 4 |
| 1.2.1 | Entradas | 5 |
| 1.2.2 | Salidas | 5 |
| 2 | Herramientas y Servicios | 7 |
| 2.1 | Requerimientos Técnicos | 7 |
| 2.2 | Requerimientos Tecnológicos | 7 |
| 2.3 | Integración Continua (CI) y Manejo de Versiones | 8 |

1 Definición del Problema

En control de calidad y el monitoreo de los procesos de producción es un aspecto de gran importancia en cualquier empresa de manufactura y ensamble.

Específicamente, empresas que fabrican juguetes de bloques de construcción deben llevar un control del inventario de piezas contenidas en cada paquete o set. Por lo tanto, pueden derivarse errores en los que las piezas de dichos paquetes no se encuentren completas o se añadan algunas de más. Por ende, se debe encontrar una forma de poder llevar un control de las piezas mediante el análisis de la línea de producción previa al empaque.

1.1. Solución a Proponer

En primer lugar, el escenario ideal debe ser una banda que recibe de una línea de producción todas las piezas requeridas para un set de juego. Las piezas deben de estar delimitadas en una grilla para facilitar su identificación. De esta manera, se pretende que una cámara perpendicular en la parte superior a la grilla capture la totalidad del escenario y pueda determinar si las piezas se encuentran completas o no; identificando de esta manera, si el set se encuentra acorde a los requerimientos específicos de cada uno.

1.2. Alcance

El alcance de nuestro producto estará orientado a intentar reconocer las diferentes piezas de un set. Sin embargo, se ha decidido dividir los alcances en diferentes categorías ue se presentarán a continuación.

1.2.1. Entradas

Las entradas estarán dadas por un set de máximo 5 piezas diferentes.

Entorno de Captura

Estas piezas en un inicio entraran al sistema mediante un captura de imagen a la banda transportadora donde se encuentran. Dicha banda, contendrá una grilla en la que se clasificarán las diferentes piezas separadas unas de otras para facilitar el proceso de reconocimiento y clasificación.

Iluminación

Se debe intentar una iluminación uniforme en toda la captura para mitigar los errores debido a este fenómeno que se puedan presentar.

Piezas

El set debe contener una cantidad de piezas considerablemente diferentes en su forma para evitar un reconocimiento incorrecto que podría afectar el buen funcionamiento del sistema.

1.2.2. Salidas

Confiables

El tratamiento de la imagen debe de arrojar que el set se encuentra en el estado óptimo y que puede ser aceptado.

No Confiables

El proceso aplicado sobre toda la imagen no permite dar un veredicto final que permita determinar si el set se encuentra completo o incompleto.

Rechazadas

El sistema debe determinar la falta de alguna de las piezas del set. Sin embargo, en caso de que se encuentren piezas de más, este también debe ser rechazado.

2 Herramientas y Servicios

2.1. Requerimientos Técnicos

Se utilizará el lenguaje C++ en el estándar C++17. Además se utilizará la biblioteca de visión artificial en tiempo real OpenCV.

Adicionalmente, se incluirán conceptos como integración continua y *testing* que permiten evaluar el progreso y estabilidad del proyecto.

2.2. Requerimientos Tecnológicos

En general, se requieren las siguientes utilidades y librerías de software:

- g++ (Arch 9.2.0) 9.2.0
- GNU gdb gbd-common-8.3-1
- Clang 9.0.1-1
- make 4.3
- CMake 3.16.4

2.3. Integración Continua (CI) y Manejo de Versiones

Se deben utilizar manejo de versiones e integración continua que permitan la verificación de calidad de los commits.

- Travis CI Ubuntu 16.04 Xenial Xerus LTS, 18.04 Bionic Beaver LTS
- git 2.23.0
- GitKraken 6.5.1
- GoogleTest