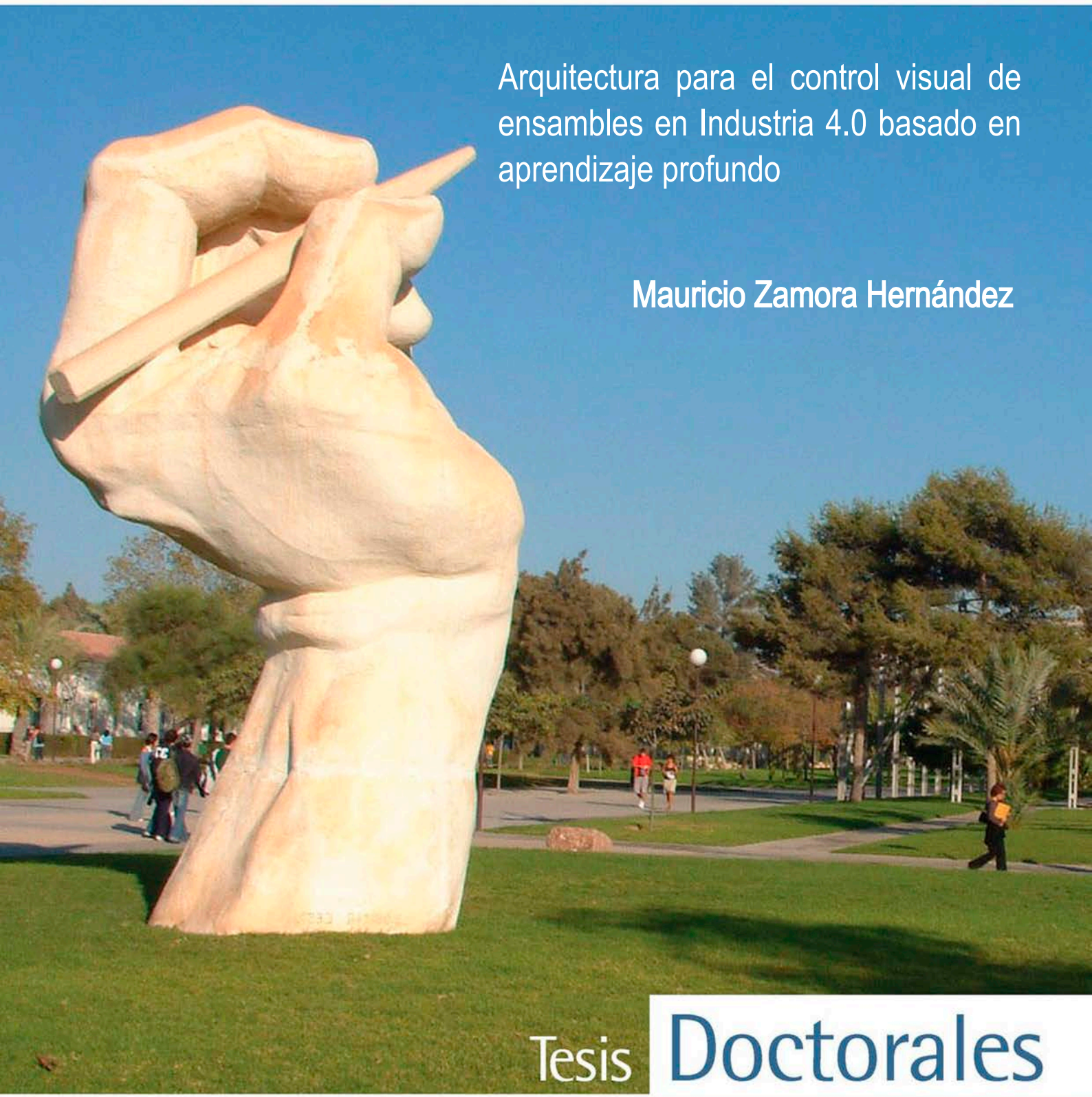
Universitat d'Alacant
Universidad de Alicante

Arquitectura para el control visual de ensambles en Industria 4.0 basado en aprendizaje profundo

Mauricio Zamora Hernández



Tesis Doctorales
UNIVERSIDAD de ALICANTE

Unitat de Digitalització UA
Unidad de Digitalización UA

Universitat d'Alacant
Universidad de Alicante

Instituto Universitario de Investigación en Informática

Escuela Politécnica Superior

Arquitectura para el control visual de ensamblajes en Industria 4.0 basado en aprendizaje profundo

Mauricio Andrés Zamora Hernández

Tesis presentada para aspirar al grado de DOCTOR POR LA UNIVERSIDAD DE ALICANTE

DOCTORADO EN INFORMÁTICA

Dirigida por:

Dr. José García Rodríguez
Dr. Jorge Azorín López

"A ningún hombre debe obligársele a hacer el trabajo que puede hacer una máquina."

Henry Ford

"Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre."

William Thomson Kelvin

"La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica."

Aristóteles

"La esperanza te hace olvidar todas las horas difíciles."

Soichiro Honda

Agradecimientos

Primero agradecer a Dios por darme fuerza para terminar este trabajo, en especial en aquellos momentos que fueron tan difíciles.

A mis padres Marcos e Isolda, mi hermana Mónica y familia que me apoyaron, con motivación para seguir.

A mis amigos de la Universidad de Costa Rica, David, Alberto y Hanzel, que brindaron su soporte para poder continuar.

Además, quiero expresar mi más profundo agradecimientos a mis tutores José García y Jorge Azorín, que han sido mi guía, me han brindado su ayuda, aliento y motivación para finalizar este largo proceso, yo los respeto como mis profesores que han sido y como amigos que han buscado el bien para mí. No quiero dejar pasar la ayuda en esta última milla de John Castro, que su aporte ha sido fundamental para finalizar este trabajo.

Alicante, 7 de julio de 2020
Mauricio Andrés Zamora Hernández

Resumen

En Costa Rica según las estadísticas de los tipos de empresas, el tipo de empresa que sobresale el mercado son las Micro, Pequeñas y Medianas Empresas (MiPyMEs), siendo este tipo de empresa un elemento diferenciador para competir en el mercado, la calidad y consistencia de los productos. Estás empresas tiene recursos económicos limitados para implementar tecnologías que le permitan incursionar en la cuarta revolución industrial. Aunque las MiPyMEs, y en concreto las del sector manufacturero, son consideradas uno de los motores económico del país, estás entidades no suelen tener beneficios del gobierno para desarrollarse digitalmente.

Esto situación generó conceptualizar un estudio para encontrar la manera en que se puede ayudar a las MiPyMEs a mejorar los procesos de manufactura, debido principalmente que en este tipo de negocios son altamente de trabajo manuales, con personal poco calificado; por lo que estos elementos afectan directamente la calidad de los productos desde su concepción. Por lo que se conceptualizó un sistema de control de calidad visual que ayude a controlar la calidad durante la fase de producción,incentivando asi el desarrollo de productos de calidad. Para esto se plantean soluciones basadas en técnicas de Visión por computadora (CV) ,junto con algoritmos de Machine Learning y arquitecturas de Deep Learning.

Este proyecto de tesis doctoral se inició realizando una revisión del estado del arte sobre los procesos de interacción de humano robots (HRI), sistemas de control automático de calidad en los sistemas de producción, la aplicación de la CV en entornos de manufactura, la utilización de arquitecturas de Deep Learning y las bases de datos de imágenes de herramientas, piezas y componentes requeridos para la manufactura. Así como bases de

datos de videos de acciones. El resultado del estudio sirvió como base para el desarrollo del estado del arte, conocer las técnicas actuales de Deep Learning que permiten identificar objetos y acciones; particularmente para el control de la producción con operarios. Por lo que se logró evidenciar que no existían soluciones inteligentes basada en visión que permitan controlar secuencias de ensamble en la producción manual.

Tras este estudio del estado del arte, se ha propuesto una arquitectura de visión por computador, reconocimiento de objetos y acciones, además de un lenguaje descriptivo. Se compone de de tres módulos principales, el primero de ellos se encarga del procesamiento visual; donde se identifican los objetos y sus ubicaciones, también se reconocen las acciones ejecutadas por el operario. El segundo módulo se encarga del procesamiento del lenguaje que describe las acciones, que va ser utilizado luego para evaluar la ejecución del operario. Y el último módulo se encarga de tomar la salidas de los dos módulos anteriores, para determinar si realmente realiza el ensamble como esta estipulado.

Además, la arquitectura es capaz de establecer cuales son las acciones siguientes que debe realizar el operario, para proveerle información de las herramientas o partes que debe tomar para continuar y minimizar los errores por uso incorrecto de herramientas o partes.

Es importante señalar que además de la arquitectura este trabajo también generan como producto, dos bases de datos. Debido a que durante la investigación del estado del arte tampoco se logró determinar la existencia de bases de datos para el entrenamiento de redes para la detección de herramientas o acciones de manufactura. La primera base de datos es de imágenes de herramientas, partes y componentes comunes de manufactura y la segunda base de datos es sobre acciones comunes en los procesos de ensamblaje.

También se propuso la generación de un lenguaje que permite describir las acciones necesarias para un proceso de ensamble. Este lenguaje que se usa para comparar con las instrucciones que se van detectando en tiempo real para determinar si el operario sigue los pasos tal y como fueron diseñados por los expertos en el diseño de productos. Este lenguaje, en conjunto con el módulo de verificación de acciones permite que el sistema genere

predicciones de instrucciones futuras.

Abstract

In Costa Rica, according to the statistics of the types of companies, the type of company that stands out in the market is Micro, Small and Medium-sized Enterprises (MiPyMEs), this type of company being a differentiating element to compete in the market, quality and consistency. of the products. These companies have limited financial resources to implement technologies that allow them to enter the fourth industrial revolution; Since MSMEs are considered as the economic engine of the country, these are entities that normally do not have government benefits to develop.

This situation generated the conceptualization of a study to find the way in which the MSMEs can be helped to improve the manufacturing processes, mainly due to the fact that in this type of business they are highly manual, with low-skilled personnel; reason why these elements directly affect the quality of the products from their conception. So a visual quality control system was conceptualized to help control quality during the production phase, to encourage the development of quality products. This requires techniques based on Computer Vision (CV), along with Machine Learning algorithms and Deep Learning architectures.

In this doctoral thesis project, a review of the state of the art on human robot interaction processes (HRI), automatic quality control systems in production systems, the application of CV in manufacturing environments began, the use of Deep Learning architectures and the image databases of tools, parts and components required for manufacturing, as well as action video databases. The study result served as the basis for the development of the state of the art, to know current Deep Learning techniques that allow identifying objects and actions; particularly for production control with

operators. So he managed to show that there was no type of solution based on Deep Learning together with Computer Vision that allows controlling assembly sequences in manual production.

Following this study of the state of the art, a computer vision architecture, object and action recognition, and a descriptive language have been proposed. They consist of three main modules, the first of which is responsible for visual processing; where objects and their locations are identified, the actions performed by the operator are also recognized. The second module is in charge of processing the language that describes the actions, which will then be used to evaluate the operator's execution. And the last module is in charge of taking the outputs of the two previous modules to determine if it actually performs the assembly as stipulated.

In addition, the architecture is capable of establishing which are the following actions that the operator must carry out, to provide him with information on the tools or parts that he must take to continue and minimize errors due to incorrect use of tools or parts.

It is important to point out that, in addition to the architecture, this work will also generate two databases as a product, because during the state of the art investigation, it was not possible to determine the existence of databases for the training of networks for the detection of tools. or manufacturing actions. The first database is of images of common manufacturing tools, parts, and components, and the second database is of common actions in assembly processes.

The generation of a language that allows describing the necessary actions for an assembly process was also carried out, this language is the input used to compare with the instructions that are detected in real time to determine if the operator follows the steps as and as designed by experts in product design. This language in conjunction with the action verification module allows the system to generate predictions of future instructions.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En esta tesis doctoral se propone una arquitectura computacional para la inspección de las operaciones que realizan los operarios en entornos de la cuarta revolución industrial (Industria 4.0) basada en visión por computador (CV) y en métodos de Deep Learning (DL). En este primer capítulo introductorio, se resume el planteamiento del trabajo de tesis doctoral incluyendo la motivación y el contexto en el que se limita este trabajo; el objetivo general marcado en la propuesta, así como los específicos del trabajo a desarrollar. Finalmente, se concluye con la propuesta de solución, que será abordada en los siguiente capítulos y las aportaciones más relevantes de la tesis.

1.1. Motivación y Contexto

La Real Academia Española (RAE) define motivación[1] como "Acción y efecto de motivar" y "Conjunto de factores internos o externos que determinan en parte las acciones de una persona". Además, la RAE también define motivar[2] como "Dar causa o motivo para algo" y "Dar o explicar la razón o motivo que se ha tenido para hacer algo".

Dada la definición anterior, procederé a detallar el porqué decidí involucrarme en esta investigación. Independientemente de que sea ingeniero de sistemas, llevo a cabo mi labor docente e investigadora en el departamento de Ingeniería Industrial (II) de la Universidad de Costa Rica (UCR). En este departamento, se están realizando propuestas de mejora en la industria, en especial en la parte de manufactura. Además, Costa Rica (CR) como país tiene el importante reto de actualizar los conocimientos del sector productivo a la Industria 4.0. Sin embargo, la investigación e innovación llevada a cabo hasta la fecha está centrada en el control visual automático de calidad de piezas o componentes de parámetros muy básicos en términos de forma, color y tamaño. Por tanto, mi motivación principal es la de dotar de inteligencia, desde mi faceta de ingeniero de sistemas, a los sistemas industriales para que avancen desde la automatización hacia un sistema flexible e integral que aborde los nuevos retos de la Industria 4.0.

Dado el contexto socio-económico en CR, las Pequeñas y Medianas Empresas (PyME) no cuentan con recursos financieros para implementar soluciones de control de calidad integrales dentro de sus procesos de producción completos, ni incluso para invertir en sistemas de inspección automática para alguna de sus fases de manufactura. Para las PyME, la calidad se considera un elemento relevante para sus estrategias comerciales ya que estas puedan atraer y retener clientes. Sirve como ventaja competitiva para la organización permitiéndoles sobrevivir en el mercado [91].

La Sociedad Americana de Calidad define la calidad como, "la totalidad de las características de un producto o servicio que tiene su capacidad para satisfacer las necesidades implícitas declaradas" [27]. En este sentido,

[1] https://dle.rae.es/motivación
[2] https://dle.rae.es/motivar

esta investigación busca asegurar el nivel de calidad en la fabricación de productos en las PyME mediante la verificación visual de que las tareas se están ejecutando según especificaciones. Esto repercute en el incremento global de los niveles de calidad, en términos de que se asegura que la ejecución de la totalidad de los procesos son acordes a la declaración de instrucciones de fabricación.

Concretamente, en CR en el año 2015, el 95 % de las empresas registradas eran micro-empresa o pequeña-empresa [9]. Según los datos del Ministerio de Economía, Industria y Comercio (MEIC) de Costa Rica, para el año 2017 las PyME aportaban el 33.41 % del total de empleos formales, que corresponden a un total de 344,390 personas [5]. Estás se concentran en un 40 % en el Gran Área Metropolitana (GAM), mientras el restante 60 % corresponde a las áreas menos desarrolladas del país, con menores índices en educación y capacitación técnica [84].

Las PyMEs, tienen la particularidad que son empresas que involucran un elevado trabajo manual en los procesos productivos. Además, con el agravante que el personal tiene un alto nivel de rotación de contratación y que están influenciados por factores como: la cultura empresarial, los valores sociales, la capacidad de aprendizaje, el cansancio, los salarios no competitivos, la alta concentración de mano de obra no calificada, entre otros [61]. Como se describió anteriormente, muchos de estos empleados son de zonas fuera del la GAM, que tiene la particularidad de producción por temporadas. En esta caso, la experiencia adquirida por los empleados aún se ve más comprometida, ya que los empleados buscan emigrar al centro del país. Provocando que las PyMEs generen una fuerte inversión de tiempo y dinero en formación de los operarios para solventar estos factores negativos que afectan directamente a la producción.

La investigación llevada a cabo en esta tesis se enmarca en los entornos de manufactura donde se utiliza un formato de celdas de producción. En estas, un operario puede producir múltiples productos y existe la posibilidad de que sean incluso distintos productos durante una jornada laboral. Además, cada operario debe configurar su entorno de trabajo según la programación de ensambles que debe realizar, para lo cual es necesario también una verificación de los elementos presente en área de trabajo [80].

Al explorar la situación anterior, me planteé posibles características que mi propuesta de investigación podría aplicar para minimizar los factores negativos descritos anteriormente, entre ellos:

1. Identificación en tiempo real de ejecución incorrecta de las instrucciones de ensamble.

2. Disminución de errores por ejecución equivocada de instrucciones de ensamble.

3. Disminución de desperdicio de materiales ya que el sistema puede indicar cuál es la herramienta correcta para la operación que va a realizar el operario, evitando que se genere daño al ensamble final por uso inapropiado de las mismas.

4. Reducción de desperdicio de tiempo por falta de herramientas, debido a que el sistema podrá indicar si se encuentran las herramientas necesarias para las secuencias de operación.

5. Reducción de tiempos de búsqueda de materiales o herramientas, al indicar al operario por medio de sistemas de alerta qué material o herramienta debe tomar.

6. Disminución de la curva de aprendizaje (para operarios novatos).

7. Reducción de tiempos, al predecir necesidades de materiales o herramientas, esto para las estaciones de producción continúa que requiere el constante llenado de los componentes por parte de un asistente.

8. Disminución de retrasos en la línea de producción ya que el sistema puede controlar el inventario de materiales, solicitando abastecimiento antes de agotarse los inventarios.

Después de un análisis detallado realicé una revisión de las características que formarían la base de la investigación, las restantes podrían ser extendidas por otros trabajos de investigación. Por tanto, se determinó que las primeras cuatro características formarían los retos principales a abordar en esta tesis doctoral.

En general, el problema que se aborda es el de supervisar en tiempo a un operario, para controlar que está ejecutando adecuadamente las instrucciones de ensamble de un determinado producto. Se trata de un reto científico que involucra diferentes áreas de investigación. En esta tesis doctoral, se plantea especificar una arquitectura de visión basada en aprendizaje máquina capaz de monitorizar cada acción del operario relacionada con el ensamble de un producto concreto. La visión por computador (CV) permitirá percibir el entorno y al operario, mientras que el aprendizaje máquina permitirá aprender a tomar decisiones en base al producto, al entorno o al operario. En base al producto, se podrá inspeccionar el mismo conforme está siendo ensamblado lo que permitirá tomar decisiones sobre el mismo antes de estar acabado (por ejemplo, determinar que ese producto no cubre los estándares de calidad. En cuanto al entorno, la arquitectura podrá determinar los elementos presentes en el área de trabajo, ya sean herramientas, partes del producto (como tornillos, arandelas, etc.) o elementos del producto final, con el objetivo de decidir si es necesario más material, las herramientas no son las adecuadas, etc. Finalmente, en cuanto al operario, la arquitectura podrá determinar qué acciones está realizando y cómo para poder tomar decisiones en cuanto a si la operación es correcta, proponer nuevas acciones o incluso, determinar el posible cansancio del operario.

1.2. Descripción del problema

Desde el punto de vista de la definición de Ingeniería Industrial dada por el Instituto de Ingenieros Industriales y de Sistemas de Estados Unidos de América, “La Ingeniería Industrial se ocupa del diseño, mejora e instalación de sistemas integrados de personas, materiales, información, equipos y energía. Se basa en el conocimiento especializado y la habilidad en las ciencias matemáticas, físicas y sociales junto con los principios y métodos de análisis y diseño de ingeniería, para especificar, predecir y evaluar los resultados que se obtendrán de dichos sistemas.” [39].

El problema abordado en esta tesis doctoral se puede circunscribir a la definición anterior. Se busca especificar una arquitectura que a través de

un mecanismo de control visual del operario en Industria 4.0 pueda verificar si un producto está siendo ensamblado según sus especificaciones. Se pretende promover el trabajo de forma integrada personas/información/equipos a través de técnicas de conocimiento especializado relacionados con la visión por computador y el aprendizaje profundo. Se persigue aplicar los principios y métodos de análisis de los estudios de trabajo para adaptar los elementos del diseño de ingeniería y mejorar la forma tradicional de trabajo. Esto afectará a la productividad de las empresas pudiendo evaluar los resultados obtenidos para aplicar rediseños de mejora continua.

La inspección automática es uno de los primeras desarrollos (que aún continua teniendo muchos retos) en la industria para la integración del trabajo de los humanos con el de las máquinas. En esta se compara un producto contra los estándares (medición de piezas o componentes) para determinar su calidad. Debido a la creciente necesidad de productos personalizados y de niveles más altos de calidad, las especificaciones de los sistemas cada vez son más estrictos. Se pretende que los sistemas puedan identificar, por sí solos, partes de las piezas que son creadas para soluciones únicas o de producción limitada, para lo cual el aprendizaje se convierte en un aspecto fundamental para llevar a cabo esta tarea. El uso de Inteligencia Artificial (IA) puede permitir que los sistemas puedan tomar decisiones en el control de la calidad en las estaciones de trabajo. Según Ferreiro and Sierra [25], estos procesos de inspección de calidad pueden ser llevado a cabo mediante sensores sencillos (peso, color, tamaño, etc.) [23] o mediante tecnologías avanzadas como la visión por computador. En esta, a través de cámaras ópticas, es posible la identificación de calidad respecto a la forma, color, así como del procesos de los productos procesados en otras estaciones de trabajo [35].

La inspección visual automática se ajusta de manera flexible a una diversa variedad de productos o procesos sin tener que invertir en nuevos componentes o sensores. Además permite la intervención de operarios interactuando con los productos y con las máquinas, debiéndose tener en cuenta los mecanismos establecidos de control e intercambio de información [49]. Las técnicas que aplican la visión por computador pueden realizar detección de colisiones [108], navegación[38] y realidad aumentada.[63].

Uno de los principales aspectos a considerar es la protección de la integridad de los operarios; se puede mejorar la interacción humano-robot utilizando aprendizaje a través de cámaras "RGB-D" para el rastreo de los movimientos de los operarios con el fin que los robots puedan predecir las intenciones y reconocer el comportamiento de los humanos con las que trabaja en forma colaborativa[115]. Esto permite crear entornos más flexibles de trabajo para la realización de las labores humanas, pero esto requiere que los algoritmos de aprendizaje automático tomen decisiones a partir de conjuntos de datos de ambientes parcialmente conocidos, teniendo que establecer políticas de control para la protección de las personas. Santoro et al. [92] proponen usar un esquema mixto, utilizando aprendizaje supervisado y no supervisado, en donde los conjuntos de datos utilizados para el entrenamiento de los patrones de comportamiento sean realizados por un "entrenador".

Este trabajo de investigación, injerencia directamente en una de las áreas de la II, el estudio del método que según Kanawaty [41] lo define como "El estudio o ingeniería de métodos es el registro y examen crítico sistemático de los modos de realizar actividades, con el fin de efectuar mejoras". Dada la propuesta de este trabajo investigación de poder evaluar las acciones de los operarios, es necesario realizar un registro minucioso de la acciones, que serán descritas a través de un análisis sistemático para conocer la forma que genera una mayor productividad en la organización.

Aunque esta propuesta de control visual para el ensamble busca impactar directamente a las PyME, se puede conmutar a empresas grandes, donde son entornos con un mayor nivel de sofisticación donde se utilizan herramientas, que provienen de entornos de *Lean manufacturing* (LM), que buscan reducir y eliminar las actividades y el desperdicio sin valor agregado [28]. Esto es según Yeen Gavin Lai et al. [121], "Cualquier elemento en la producción que no agregue valor al producto se considera una forma de desperdicio, por lo tanto, el desperdicio puede ser en forma de sobreproducción, esperando procesamiento, trabajo adicional o no conforme, inventario, movimiento y acciones correctivas.". Así que se considera que la propuesta, afecta directamente en la reducción de desperdicios, porque busca reducir o eliminar el reproceso causado por errores en las instruccio-

nes de ensamble y minimizar el producto dañado por inconsistencias en sus ensambles. *Lean manufacturing* busca explotar las nuevas tecnologías para maximizar la reducción de los desperdicios, como es el uso de la robótica, los sistemas ciberfísicos, el Big Data y el Internet of Things (IoT). Además, se pone en valor el trabajo en conjunto con los colaboradores humanos para formar un entorno común con las máquinas y trabajar de forma sinérgica en la reducción de desperdicios [35, 36, 50].

Uno de los retos actuales es considerar, en las industrias automatizadas, la interacción de los humanos con los sistemas robóticos de una forma más natural, segura, minimizando el estrés a los operarios por el uso de las tecnologías [68]. Por lo que la propuesta de investigación busca contribuir a la interacción con el operario para realizar sus labores con sistemas inteligentes.

La Cuarta Revolución Industrial (Industria 4.0) esta provocando grandes cambios en las industrias de manufactura y servicios, al involucrar a las manufacturas las técnicas de Inteligencia artificial (IA) y máquinas avanzadas. Entre las áreas donde se están realizando grandes avances, destacan la interacción de humanos con las máquinas, y la manufactura en masa de productos personalizados. Este hecho puede incrementar la complejidad de los sistemas a los operarios humanos y puede facilitar la confusión. Es imprescindible la utilización de los sistemas inteligentes para disminuir la complejidad. Esto también ha provocado nuevos retos administrativos y de gestión de la calidad: manejo de una demanda intermitente, producción esporádica, alta rotación de inventarios, la producción en lapsos prolongados, o la creación de productos nuevos constantemente. Se considera que esto generaría situaciones de estrés a los operarios, debido a que tienen que modificar la producción para adaptarse rápidamente a la demanda. Por ejemplo, los operarios más experimentados podrían confundirse durante los procesos de ensamble al estar acostumbrados a un producto anterior. También se pueden tener errores por la curva de aprendizaje a los nuevos operarios ya que estos no conocen los procedimientos de fabricación.

Un elemento crucial para poder llevar a cabo una fácil transición a la Industria 4.0 es asistir a las personas involucradas en los procesos productivos a utilizar la tecnología, especialmente el uso de la Inteligencia

artificial.

Otro de los retos relacionados está guiado por los principios de la cibernética en el estudio de los principios de sistemas complejos. En esta se involucra el control e intercambio de información entre máquinas y humanos; creando modelos que buscan la realización de un objetivo principal[96].

1.3. Objetivos de la investigación

Analizado el problema a resolver en la sección anterior, se pone de manifiesto la carencia de sistemas de inspección integrales que sean capaces de analizar el producto a lo largo de la cadena de montaje. Además, los sistemas capaces de monitorizar al operario mientras realiza su tarea carecen de interactividad. Por tanto, en esta tesis doctoral, la hipótesis de partida es que el desarrollo basado en aprendizaje máquina de un sistema de inspección visual de las acciones llevadas a cabo por un operario, durante el ensamble de un producto, permite dotar de un control de calidad adaptable, flexible y robusto que ayuda a la toma de decisiones en tiempo real en el proceso de manufactura.

Por tanto, el objetivo general de la investigación es la **especificación de una arquitectura para la monitorización y el control visual de instrucciones de ensamble para productos en el contexto de la Industria 4.0 basado en aprendizaje máquina**. Se plantea que la arquitectura sea capaz de reconocer, mediante aprendizaje, las acciones llevadas a cabo por el operador y el entorno en el que se desarrolla la misma. El entorno quedará definido tanto por las herramientas necesarias para el ensamble como por los objetos que forman parte del producto (partes del mismo, tornillos, tuercas, etc.). Para la especificación de las acciones del operario se plantea, a su vez, especificar un lenguaje que permita la descripción las mismas. Se pretende que la gestión de la producción pueda generar descripciones de montaje basadas en el lenguaje para que los sistemas basados en la arquitectura puedan realizar la inspección del ensamblado. De la misma forma, se plantea que la arquitectura sea capaz, también, de aprender el ensamblado a través de la monitorización de un experto.

Para poder cumplir este objetivo general, se plantean los siguiente objetivos específicos:

- Analizar y estudiar los trabajos relacionados que permiten la inspección visual de acciones para delimitar la frontera del conocimiento en este contexto.
- Especificar una arquitectura de visión basada en aprendizaje máquina (profundo) para la inspección de acciones del operario. La especificación requiere tanto el reconocimiento de los objetos del entorno como de las acciones llevadas a cabo.
- Definir un lenguaje de descripción visual que permita describir las acciones generales de ensamble para un operario en entornos de celdas de manufactura a partir de secuencias de imágenes.
- Definir un conjunto de datos de entrenamiento para el reconocimiento de entornos de ensamble (herramientas y piezas de uso general en la manufactura) y de acciones generales de manufactura que permitan definir los sistemas de inspección.
- Validar la propuesta de arquitectura para el control visual de ensambles en el contexto de la Industria 4.0.

1.4. Propuesta de solución

En esta tesis doctoral se propone el desarrollo de una solución dirigida al campo de la Ingeniería Industrial (II), en el contexto de la producción celdas de manufactura y que se relaciona con los subcampos de la II, tales como: diseño de trabajo y factores humanos. También incorpora la IA, especialmente la Visión por Computadora (CV) con técnicas de Deep Learning (DL) para proponer una esquema de control de operaciones de ensamble que permita corroborar que el operario sigue las instrucciones de ensamble, según la descripción del proceso.

La Industria 4.0 se ha convertido en una fuerza innovadora que ha generado una evolución acelerada en la forma de cómo las empresas realizan

sus labores cotidianas. Una de las principales actividades de una manufactura es el proceso de ensamble, que por su naturaleza el operario construye distintos productos en su jornada laboral y que podría confundir las instrucciones de fabricación.

Se propone, una arquitectura de visión basada en aprendizaje profundo capaz de controlar las operaciones de ensamble por parte de los operarios de manufactura y le asesore de pasos siguientes durante la fase de ensamble. La arquitectura, a través de la percepción visual permita corroborar que el operario sigue las instrucciones. Para realizar esta tarea, es necesario un análisis detallado los movimientos de las operaciones en la producción que permitirá optimizar los procesos.

Para realizar el análisis de movimientos, se requiere tener un registro de los movimientos esperado que debe ejecutar el operario para ensamblar un producto, por lo que se realizó una propuesta complementaria a la solución principal, que consiste en un lenguaje que permita descripción las acciones necesarias para completar un proceso de manufactura.

El aspecto de aprendizaje de la arquitectura permite que se puedan diseñar sistemas adaptados a las necesidades de la manufactura dado que podrá aprender los tipos de acciones y el entorno en el que se desarrolla. Adicionalmente, para realizar el proceso de aprendizaje es necesario establecer los conjuntos de datos para el aprendizaje. Concretamente, se plantea necesario una base de datos especializada en imágenes de herramientas, partes y componentes que pueden ser comúnmente ubicados en una celda de producción. Además, un segunda base de datos, que pueda contener vídeos con acciones básicas para los procesos de manufactura, con el fin que la propuesta pueda reconocer las mismas.

De manera general, este trabajo pretende aplicar los principios del estudio del método de trabajo mediante técnicas de visión por computador y aprendizaje máquina. Se busca analizar la interacción que existe entre los operarios y las máquinas en entornos de producción que aplican las características de la cuarta revolución industrial en la producción de productos personalizados. En este contexto, se podrán analizar los tiempos de proceso, lo que permite una planificación de la capacidad productiva en la manufactura y servicios. Además, los sistemas inteligentes podrán de-

terminar las operaciones para definir las secuencias correctas de ensamble, anticipando las necesidades de herramientas y componentes, disminuyendo el tiempo estándar, a la vez que mejorando la productividad de la organización.

En esta propuesta se considera la interacción de los humanos con los sistemas robóticos en las industrias automatizadas de una forma más natural, que involucre la seguridad ocupacional, minimizando el estrés a los operarios por el uso de los sistemas robóticos [68], en conjunto los colaboradores humanos con las máquinas forman una forma de trabajo sinérgico [35, 36, 50].

Cada operario podrá configurar y controlar su entorno de trabajo según la descripción del ensamble descrita por un lenguaje estructurado aunque se trata de un proceso complicado debido a diversos factores como los detalla Patten et al. [80]: "Los sistemas de ensamble en la fabricación están sujetos a un número creciente de variantes, tamaños de lote más pequeños y ciclos de vida más cortos". Es muy complicado para un solo operario recordar de memoria todas por las posibles combinaciones de cómo construir cada producto, componentes y sus variaciones.

La investigación tendrá como fin definir el diseño de la solución que permite controlar y evaluar las ejecuciones de los operarios en un celda de producción a través del uso de la CV y un lenguaje estructurado de descripción de operaciones de ensamble, que permite elevar los estándares de calidad en la manufactura y provea como mecanismo de sistema de sugerencias. Por ejemplo, una posible configuración de un sistema basado en la solución propuesta se puede apreciar en la figura 1.1. Aquí, se pueden identifican cuatro elementos clave:

(A) Pantalla de realimentación al operario, donde se puede apreciar la instrucción actual, las sugerencias de ensamble en caso de encontrar un error, etc.

(B) Cámara que debe estar monitorizando el área de trabajo y al operario para identificar las acciones realizadas, además de los objetos y herramientas presentes.

(C) Sistema de alerta visual, tipo semáforo, que podría codificar el signifi-

cado de la luces para alertar tanto al operario como a los supervisores de lo que esta ocurriendo durante el ensamble.

(D) Área de trabajo donde se llevan a cabo las acciones, se mantienen los componentes, piezas y herramientas. Es la zona que la cámara mantendrá enfocada.

Figura 1.1: Ejemplo de propuesta de la solución en entorno de trabajo (Fuente qualites.net y cafago.com con modificaciones)

Dada la complejidad de la propuesta, se propone una solución en múltiples componentes, que conforman la arquitectura para el control visual de ensambles. Cada una de las investigaciones y elementos clave se enumeran a continuación con una breve descripción:

- Celda de trabajo con operario. Este es el operario en su lugar de trabajo, se coloca como parte de la solución aunque es un elemento externo. Dado que es el objeto de atención de la solución, se considera relevante enumerarlo como parte de la operación de la propuesta.

- Lenguaje de Descripción de Manufactura. Se trata de un lenguaje de descripción de actividades extensible que permite definir la acciones

que la solución principal utilizará para verificar que el operario este realizando las acciones de forma correcta.

- Bases de datos de imágenes de herramientas y piezas, y base de datos de vídeos de acciones de manufactura. Se propone la generación y utilización de dos bases de datos para complementos de asistente de control visual. La primera para realizar el entrenamiento de la red neuronal capaz de identificar herramientas, piezas y componentes. Se plante generar una base de datos híbrida, compuesta de imágenes reales y sintéticas. La segunda base de datos, consta de un conjunto de vídeos, que serán grabados con las acciones descritas por el lenguaje, con el fin de poder entrenar la red para el reconocimiento de las acciones.

- Arquitectura para el control visual de ensambles en la industria 4.0 basado en aprendizaje profundo. Es el núcleo de la solución que se plantea e integra los componentes anteriores, adiciona los componentes del aprendizaje, análisis de las instrucciones, verificación de instrucciones con las acciones detectadas de forma visual y el método de recomendación.

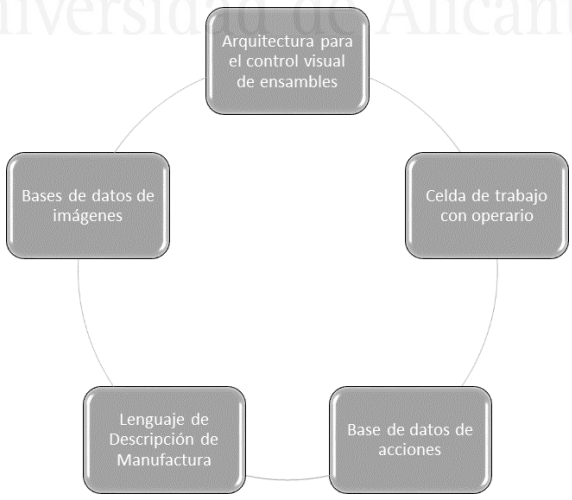


Figura 1.2: Partes de la Solución

1.5. Estructura del documento

Este documento se compone de seis capítulos. Este primer capítulo de introducción contiene la motivación y contexto que estableció la formulación de esta tesis doctoral, la definición del problema, la propuesta de la solución, los objetivos y las aportaciones de la misma.

En el capítulo 2 se realiza un análisis del estado del arte sobre los conceptos relacionados de la Ingeniería Industrial, que forman el marco conceptual del problema. En estos se incluye el estudio del método de trabajo. Además, se hace una revisión de los elementos de manufactura, de cómo son las representaciones tradicional de operaciones, cómo afecta la Inteligencia Artificial (IA) a los procesos productivos, y la Cuarta revolución industrial (Industria 4.0) y cómo esta afecta los elementos de inspección automática. Por otro lado, se revisan los métodos de aprendizaje profundo (Deep Learning, DL) y cómo se involucran las técnicas de aprendizaje de este trabajo. Por último, se realiza una revisión de las arquitecturas de DL utilizadas para los sistemas de reconocimientos y localización de objetos.

El capítulo 3, titulado "Creación de Lenguajes de Descripción de procesos de manufactura", contiene la propuesta del lenguaje diseñado para describir las instrucciones necesarias para ejecutar ensamble de productos. Dentro de este se puede apreciar la gramática del lenguaje y ejemplos de acciones de manufactura realizados con el lenguaje propuesto.

El capítulo 4, "Bases de datos de imágenes etiquetadas de herramientas y vídeos anotados de acciones de manufactura", contiene la propuesta de la base de datos de imágenes para el entrenamiento del módulo de reconocimiento del entorno de trabajo: herramientas y partes generales de los procesos de manufactura. También se detalla la base de datos de vídeos de acciones generales para la manufactura que sustentan la propuesta del lenguaje del capítulo anterior.

En el capítulo 5, llamado "Arquitectura de Control Visual de ensambles", se explica la arquitectura propuesta para el sistema de control visual de ensambles. Se explican los elementos relevantes que lo componen: el módulo de procesamiento visual, el módulo de procesamiento del lenguaje de descripción de manufactura y el módulo de procesamiento de comandos

que orquesta los módulos anteriores. El módulo de procesamiento visual está conformado por los sistemas de detección de objetos y acciones. El módulo de procesamiento del lenguaje tiene como elemento principal el intérprete del lenguaje.

Por último, en el capítulo 6 se presentan las conclusiones de este trabajo de tesis doctoral. Se explican los trabajos futuros que se pueden generar en un corto, medio y largo plazo sobre los temas expuestos; además de las publicaciones consecuencia de esta investigación.

Capítulo 2

Estado del Arte

Este segundo capítulo se presenta una revisión del estado del arte general y transversal a la tesis. Se hace un resumen de la creación de nuevas industrias, productos y servicios con los avances de la Cuarta Revolución Industrial; la interacción humano-robot que incluye el machine learning y la visión por computadora. Siendo elementos a considerar debido a que permiten obtener entornos colaborativos entre personas y robots. También se exponen los conceptos que describen el ámbito de la Ingeniería Industrial que definen el marco de acción de la investigación. Además, se estudian los trabajos relacionados con Deep Learning (DL) y Visión por Computadora (CV), dando prioridad aquellos trabajos que permiten establecer la definición de arquitecturas que permitan resolver el problema de la tesis. Además, se revisan propuestas que plantean la necesidad de la utilización de la inteligencia artificial en todos los procesos de la industria 4.0 como elemento crucial de enlace los operarios y los sistemas inteligentes. Para concluir se hace un resumen de la arquitecturas de DL para el reconocimiento y localización de objetos.

2.1. Ingeniería Industrial (II)

En esta sección revisamos conceptos básicos de la Industria 4.0 y las áreas particulares en que se delimita el contexto de la solución que se propone en esta tesis doctoral.

2.1.1. Definición de Ingeniería Industrial

Según el Instituto de Ingenieros Industriales y Sistemas (IIIS), con base en Estados Unidos de América, se resume el concepto de Ingeniería Industrial en su cuerpo de conocimiento como:

> "La Ingeniería Industrial se ocupa del diseño, mejora e instalación de sistemas integrados de personas, materiales, información, equipos y energía. Se basa en el conocimiento especializado y la habilidad en las ciencias matemáticas, físicas y sociales junto con los principios y métodos de análisis y diseño de ingeniería, para especificar, predecir y evaluar los resultados que se obtendrán de dichos sistemas." (Institute of Industrial and Systems Engineers, 2019)

En el cuerpo de conocimiento del IIIS [39], se establece una división de las principales áreas en que se enfoca la Ingeniería Industrial, el presente trabajo impacta en varias áreas, que se detallarán posteriormente:

- Diseño del trabajo y medición
- Operaciones, investigación y análisis
- Ingeniería del análisis económico
- Ingeniería de instalaciones y gestión de la energía
- Ingeniería de calidad y fiabilidad
- Ergonomía y factores humanos
- Ingeniería y gestión de Operaciones
- Gestión de la cadena de suministro

- Ingeniería para la administración
- Seguridad
- Ingeniería de Información
- Ingeniería de diseño y manufactura
- Diseño y desarrollo de productos
- Ingeniería y diseño de sistemas

2.1.2. Diseño del trabajo y medición

Diseño de trabajo y las mediciones comprende las herramientas y técnicas que son usadas para establecer el tiempo promedio que un trabajador emplea para llevar a cabo una tarea específica, en un nivel definido de rendimiento y en un entorno de trabajo definido. Este análisis permite crear entornos de trabajo estandarizados que permiten maximizar la satisfacción del trabajador y crear el mayor valor posible para la empresa y sus clientes [39].

2.1.3. El estudio del método

Un tema que forma parte de las bases de la Ingeniería Industrial es el estudio del método, que Kanawaty [41] define como "El estudio o ingeniería de métodos es el registro y examen crítico sistemático de los modos de realizar actividades, con el fin de efectuar mejoras".

El estudio del método permite estudiar los procesos desde sus elementos básicos, como los movimientos necesarios para completar tareas. De este modo, se pueden realizar mejoras en los procesos productivos, ya que se podría determinar cambios en las secuencias o reducción de movimientos innecesarios.

Con estos análisis de operaciones se pueden establecer cálculos con alto nivel de precisión en los tiempos estándar. Esto a su vez permite hacer estimaciones precisas de la capacidad y producción, crucial para la planificación y toma de las decisiones en la manufactura o servicios.

Este estudio se representa normalmente de una forma gráfica, llamado flujograma de operaciones, tal como se puede apreciar en la figura 2.1, esta figura es para un proceso regular, también existen versiones para cada una de las manos.

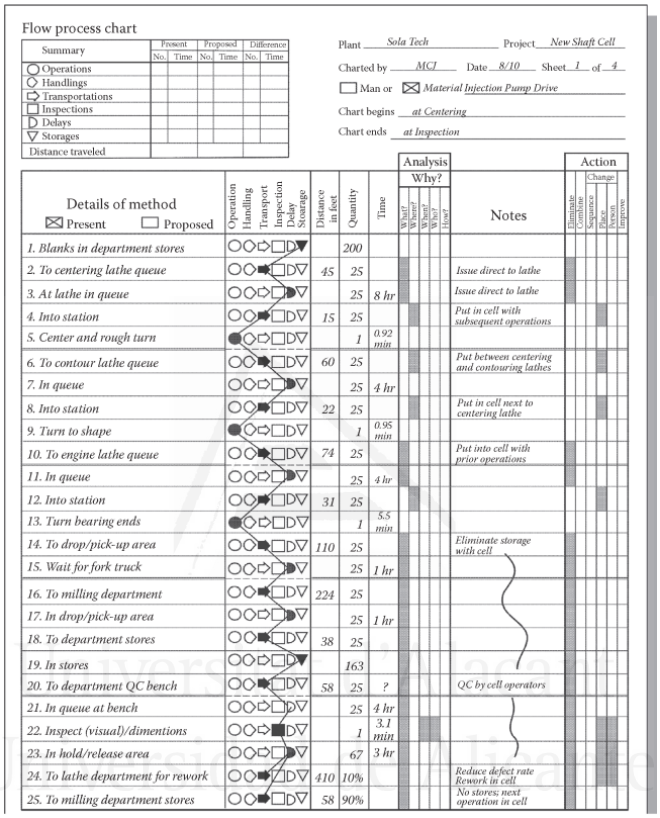


Figura 2.1: Ejemplo de un flujograma de operaciones (Fuente ebrary.net)

2.2. Retos actuales en la manufactura

La manufactura a nivel mundial está expuesta a una alta competitividad en el lanzamiento de nuevos productos, por lo que la ayuda de la inteligencia artificial puede proveer mecanismos para ayudar la planificación y diseño de nuevos productos [18]. Para poder llevar a cabo esto, se debe optimizar cada uno de los procesos de la ingeniería de producción; desde las fases de diseño [83], planificación de procesos, cálculos complejos,

modificaciones de celdas de producción [18].

En los procesos de producción se generan y consumen altos volúmenes de datos; desde el inicio con la definición de necesidades por parte de los clientes, pasando a las siguientes etapas como diseño y planificación de la solución, o la programación de las máquinas y robots con la secuencia de producción de los componentes del producto. La unión de todos estos elementos genera un sistema muy complejo para su control y monitoreo, tarea muy compleja para un ser humano en la toma de decisiones, por lo que es conveniente implementar un sistema de inteligencia artificial que facilite dicha tarea [83]. Al integrar, además de las máquinas, el factor humano, se agrega un nivel adicional de complejidad para las personas que toman las decisiones y aquellas que tiene que programar las configuraciones de los equipos. Por esta razón, Monostori [71] recomienda usar técnicas de reconocimiento de patrones, sistemas expertos, redes neurales artificiales o sistemas difusos, generando sistemas mixtos entre técnicas de manufactura tradicionales e Inteligencia Artificial [71].

Otra consideración al utilizar robots y maquinaria junto con personas en entornos productivos, es proteger la integridad física de los operarios. Para lo cual Cherubini et al. [13] recomiendan utilizar herramientas asistidas con visión por computador que facilitan controlar la cercanía entre ambos elementos. Pero no solo se debe usar la visión por computador como un simple sensor de distancia. En [93], la visión por computador es un elemento clave para determinar lar rutas de movimiento de los actuadores, tales como los brazos robóticos. Sistemas para evitar colisiones con las personas u otros equipos, que pueden incluir inteligencia artificial, pueden prevenir accidentes o daños durante sus labores de producción [93].

Herrero et al. [37] argumentan que la visión por computador no solo permite generar acciones preventivas, sino mejorar la HRI al utilizar comandos basados en habilidades visuales que faciliten la manipulación de los robots por parte de las personas. En Ding et al. [16] se propone no solo buscar algoritmos más elaborados para definir patrones para la manipulación de los robots, sino también acompañar las visión por computador de una base de datos de elementos visuales que faciliten la interpretación del entorno para mejorar el aprendizaje y reconocimiento de figuras y objetos.

Con el fin de poder establecer reacciones más naturales en acciones de los robots.

Un aspecto importante a considerar es el control de los movimientos de los robots para evitar golpear objetos o personas, inclusive aquellos que no fueron usados en el entrenamiento, utilizando algoritmos que permiten predecir colisiones potenciales [4]. Para poder llegar a estos niveles de predicción de posibles colisiones es necesario que los robots puedan aprender a través de sistemas de reconocimiento visual, en entornos de la vida cotidiana o en condiciones especiales de iluminación que permita identificar a las personas y objetos [30].

También es importante investigar cómo mejorar el nivel de precisión entre la detección de rostros humanos y personas en tiempo real ajustándose a distintos ambientes [66]. No solo es importante que los robots puedan distinguir personas o rostros humanos y objetos, sino también gestos, Ya que la interacción humano-humano es algo natural, que va a ser trasladado a los robots, por lo que ya hay avances importantes en este área [10].

2.3. Entornos Lean Manufacturing para diseño y especificación de productos

En esta sección se hará una revisión de los elementos utilizados en la manufactura bajo el paradigma de Lean Manufacturing: el diseño de productos en entornos dinámicos, organización la información y especificación de los pasos a seguir para la manufactura un producto. Además, se revisa la propuesta de Taiichi Ohno, uno de los fundadores de esta filosofía de trabajo.

> "Para que una persona encargada de producción sea capaz de escribir una hoja de instrucciones que otros empleados puedan entender, debe estar muy convencido de su importancia."
>
> –Taiichi Ohno

2.3.1. Diseño

En los entornos tradicionales de manufactura, los diseñadores creaban un producto que se creaba masivamente. En la actualidad, tanto los diseñadores de la empresa como los usuarios finales pueden solicitar la fabricación de productos a través de modelos creados por aplicaciones Computer-aided design (CAD) [29]. Esto genera un alto nivel de estrés en empresas de manufactura, ya que deben reiniciar los procesos de producción para un nuevo producto, que además podría no ser para producción en masa, lo que aumenta sus costos operativos.

Un elemento primordial en la manufactura, tras la fase diseño de un nuevo producto, es la generación de un programa piloto de producción que proporcione información para crear el modelo usado en la manufactura en masa. Esta ejecución inicial de fabricación genera conjuntos de datos pequeños. Tsai y Li [104] proponen usar Artificial Neural Networks (ANN) con función de aprendizaje Diffusion-Neural Network (DNN) para optimizar los modelos iniciales de producción.

2.3.2. Organización de la información

Los puntos claves para el diseño de un proceso de producción son: la definición de la demanda y las necesidades particulares de cada producto. Una novedosa técnica, conocida como manufactura social [52], es utilizada en la actualidad para aprovechar la información cruzada de las empresas empleando las ventajas de las tecnologías de información y comunicaciones. Además, se obtiene información del Internet de las cosas, Big Data, tecnologías de la nube y procesos de manufactura avanzada [12].

Los entornos de manufactura social y de manufactura en la nube generan grandes volúmenes de datos no estructurados. Dichos datos no son adecuados para procesamientos tipo minería de datos, por lo que se han realizado avances es la extracción de información utilizando redes neuronales basadas en casos y reglas [52].

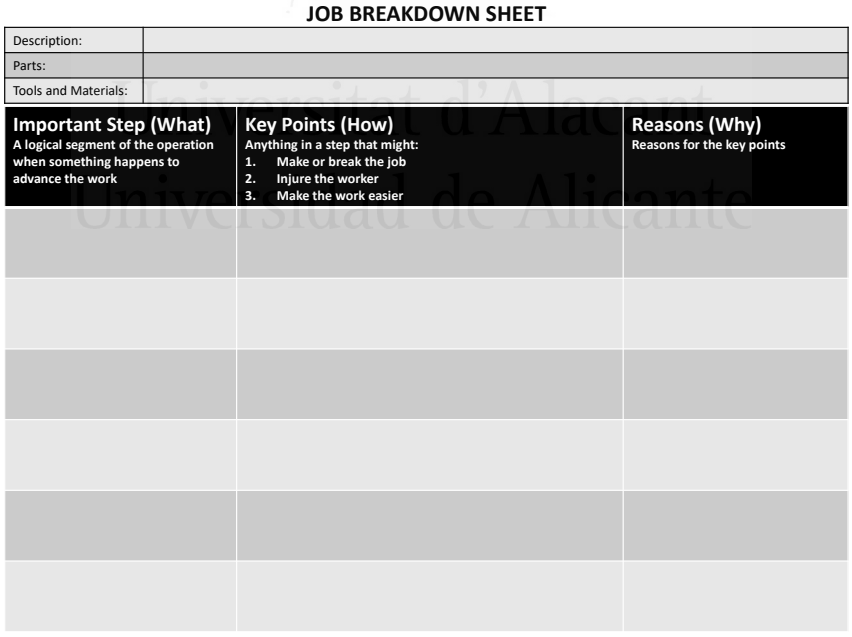
Se está utilizandoMachine Learning (ML) en la programación dinámica, específicamente con las técnicas de Redes neuronales de retro propagación y razonamiento basado en casos (CBR), con el fin de establecer las re-

glas para iniciar la producción y disminuir los tiempos necesarios para la producción [82].

Dentro de la filosofía de "Lean Manufacturing" se plantea una forma de representar las instrucciones sobre como realizar una tarea. En este caso particular, nos interesa el ensamblaje de un producto usando herramientas.

Las instrucciones proporcionadas en este enfoque servirán no solo para el entrenamiento de nuevos empleados, sino también para enseñar habilidades nuevas o mejorar las operaciones existentes en procesos conocidos por operarios regulares [110].

La hoja de desglose de trabajo (JBS), es un formato estructurado para captura del conocimiento necesario para e desarrollo de un trabajo. Es un requisito para la preparación formal del entrenamiento y registro del conocimiento del trabajo [110]. En la figura 2.2 se puede apreciar un ejemplo de un JBS.



JOB BREAKDOWN SHEET

Description:	
Parts:	
Tools and Materials:	

Important Step (What) A logical segment of the operation when something happens to advance the work	**Key Points (How)** Anything in a step that might: 1. Make or break the job 2. Injure the worker 3. Make the work easier	**Reasons (Why)** Reasons for the key points

Figura 2.2: Ejemplo de un Job Breakdown Sheet (Fuente lean.org)

2.4. Entrenamiento en entornos productivos

En otras disciplinas, tales como la Ingeniería de Software, el factor de la capacitación es crucial. Para desarrollar este tipo de capacitaciones se consideran dos puntos críticos [76]:

1. Necesidad de identificar las habilidades que debe tener el personal.
2. Mantener actualizado el programa de capacitación para adaptarse a los rápidos cambios en las técnicas y tecnologías.

Ambos requisitos se consideran necesarios para el desarrollo de esta investigación. Ya que en la industria se tienen las mismas necesidades, incluso más marcadas, debido a que se tienen muchos puestos que son de demanda rotativa según las necesidades de los análisis de planificación de la producción. Lo que nos motiva a buscar soluciones para mejorar las técnicas de capacitación del personal sin pérdida de calidad.

En la cuarta revolución industrial se logra cambiar el modelo de producción, al conectar tecnologías como el cloud, big data, analytics y el internet de las cosas, permitiendo la interacción en tiempo real dentro de la misma industria, así como la de los clientes con las empresas productoras [1, 124].

También se presenta la necesidad de mejorar las herramientas de producción, para adaptarlas a nuevos retos en la producción, en especial de como los operarios deben construir sus productos. Esto se debe a que ha evolucionado el proceso de diseño, manufactura y reciclaje, que afecta los ciclos de vida de los productos. Además de las nuevas tendencias y cambios en la creación de productos personalizados que requieren que los operarios mejoren o aprendan nuevas habilidades de ensamble.

Otro elemento a considerar, es la tasa de accidentes en la industria, que varía según el tipo particular de industria. Entre las más altas se encuentra la de construcción, donde Melzner [69] propone utilizar listas de verificación y descripciones manuales para identificar y disminuir los accidentes.

Esta propuesta puede ser adoptada para aplicarse a otras industrias donde se requieran secuencias de producción. En especial en aquellas organizaciones donde los métodos de producción han evolucionado en esquemas

de producción modular, que permiten satisfacer las necesidades individuales de los usuarios- Esto puede generar confusión en los operarios al crear una gran gama de productos diferentes con características comunes.

El método de producción ha sido empleado desde la producción en masa temprana, pasando por la producción modular posterior, hasta la producción inteligente actual. El desarrollo de la tecnología ha permitido satisfacer las necesidades individuales de los usuarios [124].

2.5. Aplicaciones de la tecnología

De entre los temas tratados de la Cuarta Revolución Industrial, hacemos énfasis en los sistemas cibernéticos de manufactura, también conocidos como "cibermanufacturing". Estos sistemas plantean la interconexión fabril de los distintos elementos humanos con sistemas automatizados complejos, involucrando los sistemas computacionales de control y el intercambio de información de las operaciones de fabricación y los sistemas robóticos [96].

También se utiliza la inteligencia artificial para crear modelos de trabajo que solventen la toma de decisiones y anticipación de situaciones problemáticas en el flujo de producción [96]. En este tipo de entornos lo que se buscaría es que la interconexión entre humanos y robots sea lo más natural posible. Ya que, según [68], se pueden generar entornos que minimicen el estrés de los operarios al usar sistemas robóticos complejos.

En este tipo de entornos, tal como se menciona en Hedelind and Jackson [35], Hermann et al. [36], Lee et al. [50], se refuerza el concepto de automatización e intercambio de datos como núcleo en las tecnologías de manufactura. Donde las tecnologías como la robótica, sistemas ciberfísicos, Big Data y el Internet de las cosas, son la base en la construcción de entornos colaborativos con las personas.

Al aumentar la complejidad de los sistemas, el principal elemento que hay que considerar para la construcción de estos nuevos entornos integrados de producción son los humanos. Estos podrán hacer uso de tecnologías de interacción con robots y máquinas, como es el caso de la Realidad Aumentada (AR). En [68], se plantea el ejemplo de una planta de energía térmica en Bosnia-Herzegovina. Para evitar que los operarios cometan

equivocaciones, y por ende proteger su integridad física, se hace uso de dispositivos móviles que integran sistemas de AR. Esto les facilita utilizar listas de chequeo en tiempo real. Casos como estos se pueden encontrar recurrentemente en otras investigaciones.

Como se mencionó en la sección anterior, la visión por computador es un elemento importante a considerar en la nueva era de manufactura. Ya que sus aplicaciones en interacción humano-robot pueden ser aplicadas, por ejemplo, en el control de calidad en la manufactura, detección de colisiones [108], navegación [38] y realidad aumentada [63]. Sin embargo, para poder aplicar técnicas de aprendizaje automático en estos contexto, se requiere la intervención de operarios para poder entrenar la inteligencia artificial utilizando los robots [49]. Para esto es importante tener mecanismos establecidos de control e intercambio de información para asegurar mecanismos de calidad.

La CV también se usa en los procesos de inspección automática, donde se utilizan técnicas de aprendizaje supervisado. Por ejemplo, como señala Ferreiro and Sierra [25] se puede usar en los procesos industriales donde se busca controlar la calidad en las estaciones de trabajo [25]. Estos procesos de inspección de calidad pueden emplear sensores sencillos para captar peso, color o tamaño [23]. Además, se puede evaluar la calidad respecto al aspecto de los productos procesados en las estaciones de trabajo [35].

La visión por computador se puede usar para ayudar a proteger la integridad de los operarios. Como describen Xiao et al. [115] una herramienta para mejorar la HRI son las cámaras RGB-D que permiten el rastreo de los movimientos de los operarios para que los robots puedan predecir las intenciones y reconocer el comportamiento de las personas con las que colaboran [115].

Esto permite crear entornos más flexibles de trabajo para la realización de las labores humanas. Por otra parte, se requiere que los algoritmos de aprendizaje automático tomen decisiones a partir de conjuntos de datos de ambientes complejos. En estos entornos, pueden emplearse un entrenamiento supervisado, o por limitaciones de datos, utilizar entrenamiento no supervisado. Debido a ello, Santoro et al. [92] proponen usar un esquema mixto, utilizando aprendizaje supervisado y no supervisado en donde los

conjuntos de datos utilizados para el entrenamiento sean obtenidos en base a los patrones de comportamiento obtenidos por un "entrenador" [92].

Las técnicas de obtención de información a través de patrones también son mencionadas por Ericson, Gary; Franks, Larry; Rohrer [19], como una poderosa herramienta en los ambientes productivos en los que encontramos gran cantidad de contextos. En estos ambientes no se conocen las relaciones entre las entradas y salidas, por lo que es necesario utilizar algoritmos que permitan organizar la información o encontrar patrones [19].

2.6. La cuarta revolución industrial

La cuarta revolución industrial (Industria 4.0) no es solo un tema de moda. Consiste en la definición de las nuevas normas a las que la industria actual debe adaptarse. Dado que el concepto abarca una amplia variedad de áreas de trabajo se realizó esta investigación que resume las ideas y hallazgos de otros investigadores en este campo.

Este trabajo hace una sinopsis de los argumentos que giran en torno a la integración de la robótica con los operarios en las ciber manufacturas para la mejora de productividad, apoyados en el uso de las tecnologías como la visión por computadora, aprendizaje automático e interacción humano-robot, que a la vez permitan crear entornos colaborativos y seguros para el humano y los sistemas automatizados.

La industria 4.0 esta provocando una actualización en las empresas a través de una transformación en sus procesos productivos. Donde la automatización de sus procesos e intercambio de datos son el núcleo en las tecnologías de manufactura, en conjunto con los colaboradores humanos que conforman un entorno común con las máquinas para trabajar de forma sinérgica [35, 36, 50].

En la cuarta revolución industrial, el concepto de automatización inteligente es uno de sus ejes principales. Estos entornos productivos deben considerar la robótica como base de sus tecnologías. Así como, la Inteligencia Artificial, Sistemas ciberfísicos, Big Data, e Internet de las cosas en entornos colaborativos de Robots y personas, como una unidad.

Con el gran nivel de competencia mundial, la manufactura tiene que

estar bien planificada para responder rápidamente con productos de alta calidad [18]. Para poder llevar esto a cabo, se debe ejercer presión a cada uno de los procesos de la ingeniería de producción; desde las fases de diseño [83], planificación de procesos, cálculos complejos y modificaciones de celdas de producción [18].

Existe la necesidad de mejorar las herramientas de producción, para adaptarlas a nuevos retos en la producción, las nuevas tendencias y cambios en la creación de productos personalizados requieren que los operarios mejoren o aprendan nuevas habilidades de ensamble, que son soportadas con máquinas inteligentes [1, 124].

Además, en esta sección se explica el concepto de cibermanufactura, uno de los pilares de la Industria 4.0, y la inspección automática que se realiza en la actualidad en el contexto la cuarta revolución industrial.

2.6.1. Ciber manufactura

La Industria 4.0 se caracteriza porque los procesos de manufactura están vinculados con la personalización de la producción, lo que aumenta la complejidad de la fabricación, donde la comunicación y el control son pilares esenciales. Las herramientas para materializarlo son la robótica cognitiva, big data, IoT y lo modelos predictivos de toma de decisiones [17] [42].

2.6.2. Inspección Automática

Una de las técnicas comúnmente utilizadas en la automatización es la inspección automática. Se basa en una comparación contra estándares de medición de piezas o componentes. Debido a la creciente necesidad de productos personalizados y de niveles más altos de calidad, se han definido nuevas necesidades, donde se busca que los sistemas puedan aprender a identificar piezas que son creadas para soluciones únicas o de producción limitada. El aprendizaje supervisado es la alternativa que va a permitir integrar al sistema la toma de decisiones en el control de la calidad dentro de las estaciones de trabajo [25]. Estos procesos de inspección de calidad pueden llevarse a cabo tanto por sensores sencillos como de peso, color,

tamaño [23], como por técnicas de visión por computador, usando cámaras ópticas que permiten la identificación de calidad respecto al aspecto de productos procesados en otras estaciones de trabajo [35].

En el caso particular de la CV, su uso se ajusta a una diversa variedad de productos o procesos sin tener que invertir en nuevos componentes, ya que, basta con volver a entrenar el sistema con datos conocidos. En este proceso es necesaria la intervención de operarios interactuando entre ellos y con las máquinas. Para ello se deben establecer mecanismos de control e intercambio de información [49]. La CV es usada también en la manufactura para el control de calidad, detección de colisiones [108], navegación [38] y realidad aumentada [63].

En los ambientes productivos existen muchos contextos en los cuales se tiene información de la que no se conoce las relaciones entre sus entradas y salidas, donde se necesita algoritmos que permitan organizar la información o encontrar patrones [19].

Uno de los principales puntos a considerar es la protección de la integridad de los operarios. Para esto se está mejorando la interacción humano-robot utilizando aprendizaje a través de cámaras "RGB-D" para el rastreo de los movimientos de los operarios. De manera que los robots puedan predecir las intenciones y reconocer el comportamiento de las personas con las que colabora [115].

Esto permite crear entornos más flexibles de trabajo para la realización de las labores humanas. Pero también requiere que los algoritmos de aprendizaje automático tomen decisiones a partir de conjuntos de datos de ambientes parcialmente conocidos, teniendo que establecer políticas de control para protección de las personas. En [92] proponen usar un esquema mixto, utilizando aprendizaje supervisado y no supervisado. En donde los conjuntos de datos utilizados para el entrenamiento de los patrones de comportamiento son proporcionados por un "entrenador".

2.7. Interacción

En la industria se busca crear entornos de trabajo donde la colaboración entre humano y robots sea cada vez mayor, generando una serie de nuevos

retos. Por ejemplo, la protección de la integridad de las personas. Para ello, los mecanismos de comunicación entre humanos y robots tienen que ser más seguros [32].

Un factor que incrementada la complejidad es el ruido producido por la maquinaria de las fábricas que deteriora la interacción verbal. Un punto importante a considerar es la seguridad ocupacional. En especial, en los entornos donde hay operadores jóvenes y sin experiencia que usualmente, no leen con cuidado, no acatan las normas de seguridad y no siguen las instrucciones para evitar lesiones [103].

Entre los distintos tipos de robots que se pueden encontrar en un entorno industrial, se encuentran los robots móviles, robots fijos y recientemente robots sociales. Los robots móviles sirven para desplazar a través de la planta elementos como: materia prima, suministros, o producto terminado entre otros. En la actualidad siguen rutas predefinidas como marcas con líneas de colores. Los robots reconocen estas líneas a través de sensores ópticos o cámaras.

Con un esquema de producción más dinámico, la interacción de humanos con robots es mayor al compartir un mismo espacio físico. Wang et al. [107] proponen un modelo de planta 3D, enlazando en tiempo real sensores de movimiento para imitar los modelos con los elementos de la realidad y calcular la distancia mínima entre humanos y robots, generando un sistema activo de detección de colisiones [107].

En un entorno industrial se puede mejorar la HRI utilizando visión por computador y realidad aumentada. La visión por computador permite ayudar en la auto localización y el mapeo (SLAM), detección y rastreo de personas, identificación de actividades humanas, o de expresiones faciales. Se puede sobreponer el contenido digital en imágenes del entorno en dispositivos móviles para crear sistemas de realidad aumentada que ayuden a los operarios a interactuar en tiempo real.

Adicionalmente, se pueden utilizar dispositivos “wearable”, que faciliten las actividades humanas usando visión de primera persona (FPV) [53].

En la Tabla 2.7, se puede apreciar que la mayor parte de la interacción encontrada en los artículos científicos consultados fue habitualmente

Tabla 2.1: Aplicaciones del uso de CV en actividades industriales

Ref.	Presentación	Intercambio de Inf.	Autonomía (LOA)	Applications
Tatic and Tesic [103]	Móvil	Visual	Decisión computacional reducida	Detección eventos
Lee et al. [50]	Computadora	Visual	Decisión humana completa	Procesos de control Navigación Detección eventos Interacción
Wang et al. [108]	Móvil	Touch	Decisión humana completa	Inspección automática Navigación, Detección eventos, Interacción,
Leo et al. [53]	Móvil	Touch	Decisión computacional reducida	Inspección automática
Mehlmann et al. [67]	Computadora	Voz	Decisión humana completa	Procesos de control
Santoro et al. [92]	Móvil	Visual	Decisión computacional reducida	Interacción
Meisner et al. [68]	Móvil	Voz	Decisión humana reducida	Navigación

a través de mecanismos móviles, especialmente tabletas, en donde, la interacción se realizaba a través de botones y mostrando información gráfica sobre los procesos, además de estadísticas de uso de los dispositivos (tanto máquinas inteligentes como convencionales), esto con el fin de facilitar el desplazamiento de las personas. En cuanto al nivel de autonomía, como

Tabla 2.2: Interacción Humano Robot

Ref.	Aplicaciones	Aplicaciones	Tipos de algoritmos
Monostori [71]	Manufactura	Red neuronal	Supervisado, No supervisado
Leng and Jiang [52]	Manufactura	Red neuronal	Supervisado, No supervisado
Ferreiro and Sierra [25]	Manufactura	Red neuronal	Supervisado, No supervisado
Tatic and Tesic [103]	Bioinformática	Clustering	Supervisado
Priore et al. [82]	Manufactura	Red neuronal	Supervisado
Santoro et al. [92]	Robótica Social	Red neuronal	Supervisado, No supervisado
Meisner et al. [68]	Navegación	Red neuronal	Supervisado
Tsai and Li [104]	Manufactura	Bootstrapping	Supervisado
Xiao et al. [115]	Interaction	Non parametric	Supervisado
Rani et al. [86]	Bioinformática	Support vector machines	No supervisado
Panait et al. [78]	Robótica cooperativa	Red neuronal	Supervisado, No supervisado
Mohammad and Nishida [70]	Robótica Social	Red neuronal	Supervisado
Ramík et al. [85]	Robótica Cognitiva	Algoritmos Genéticos	Supervisado
Vlassis et al. [106]	Navegación	Algoritmos Genéticos	Supervisado
Hornung et al. [38]	Navegación	Red neuronal	Supervisado, No supervisado
Guo et al. [34]	Optimización and metaheurística	Red neuronal	No supervisado
Li and Yeh [54]	Optimización and metaheurística	Red neuronal	Supervisado
Lee and Ha [51]	Percepción	Red neuronal	Supervisado
Sudha et al. [100]	Optimización and metaheurística	Red neuronal	Supervisado

se muestra en la Tabla 2.7, los principales dispositivos para realizar la interacción son aquellos controlados directamente por humanos. Muy pocos

equipos proveen capacidades de control total por inteligencia artificial o donde la inteligencia artificial controla la mayor parte de las decisiones.

Muchas de las capacidades de decisión automática se están utilizando para determinar eventos (relacionado a estímulos sobre cada máquina, no se están controlando eventos generados por procesos de interacción con otras máquinas inteligentes en procesos complejos). Además, se utiliza la capacidad de aprendizaje automático en el reconocimiento de patrones para la determinación de navegación en ambientes controlados.

En cuanto a la interacción verbal o visual, estas son formas que tienen mayor relación con el aprendizaje automático según los datos identificados en la Tabla 2.7. Estos mecanismos buscan una interacción más natural con las personas. En el caso de la forma visual, la mayor parte de los casos son sistemas de visión por computador, en donde se está utilizando para crear sistemas de realidad aumentada para facilitar la interacción. Al usar sistemas de proyección ocular o a través de dispositivos móviles que proveen información adicional sobre el ambiente de trabajo.

El análisis de la Tabla 2.7, es una recopilación de los artículos evaluados, en donde se identifica que el enfoque de mayor uso en el aprendizaje automático son las redes neuronales con un 68 % de las ocurrencias identificadas en la investigación realizada. Junto con la técnica de entrenamiento supervisado que está presente en un 88 % de las ocasiones. Sin embargo, a pesar de su uso tan intensivo siguen existiendo retos abiertos, ya que, sus aplicaciones son muy amplias, entre ellas: manufactura, navegación, optimización y metaheurística, bioinformática, interacción, robótica colaborativa, cognitiva y social. Por último la percepción por computador, que integran el 80 % de las aplicaciones identificadas.

Los sistemas inteligentes serán la base sobre la cual se sustentará la nueva industria. Un entrenamiento adecuado de las redes neuronales es vital en las nuevas aplicaciones industriales. Ya que se busca que la interacción de los robots y operarios sea lo más natural posible. Además, se pretende que colaboren con las decisiones automáticas en los procesos productivos.

Aunque la nueva industria se enfocará en sistemas inteligentes, la interacción con las personas siempre estará presente en algún punto del proceso global. Se puede apreciar que la mayor parte de la interacción se realiza

obteniendo información mediante sensores, tal como se puede apreciar en la Tabla 2.7.

Se busca establecer una adecuada interconexión, más allá del uso de sensores para las micro, pequeñas y medianas empresas (MiPyME). El uso combinado de maquinaria tradicional y operarios con supervisión y monitoreo en tiempo real por inteligencia artificial permitirá maximizar la producción, mejorar la interacción humano robot y minimizar los riesgos de salud ocupacional al personal y los riesgos físicos a los elementos productivos.

2.8. Aprendizaje automático y aprendizaje profundo

Este trabajo doctoral tiene su base en la utilización de técnicas y arquitecturas que provienen del aprendizaje automático (del inglés Machine learning (ML)) y el aprendizaje profundo (del inglés Deep learning (DL)), por lo que a continuación se hacen una introducción a estos conceptos.

2.8.1. Aprendizaje automático - Machine Learning (ML)

El aprendizaje automático (ML) es una rama de la Inteligencia Artificial (IA) que involucra algoritmos que proveen a los sistemas computaciones la capacidad de inferir patrones desde los datos [3].

Pero para esto se requieren técnicas para poder representar este conocimiento, estrategias generales de como el sistema va a aprender. Donde el tipo de problemática a resolver influye en la selección de las variables a utilizar. Para esta tarea usamos el aprendizaje de características.

El aprendizaje de características es el conjunto de métodos que permite a la máquina descubrir de forma automática las características (representaciones) necesarias para la clasificación o detección de los datos. [3] Este proceso de aprendizaje se logra, dependiendo del problema, mediante tres enfoques [101]:

2.8.1.1. Aprendizaje supervisado

Este tipo de aprendizaje ofrece la posibilidad de aprender desde un conjunto de datos categorizados o etiquetados, proveyendo un modelo predictivo capaz de representar y generalizar un patrón de comportamiento de los datos. Una vez que el modelo es creado, este es capaz de clasificar y categorizar los nuevos casos del problema que está tratando de resolver [26].

2.8.1.2. Aprendizaje no supervisado

Comienza con un conjunto de datos que no tienen categorías o etiquetas. Estos datos son analizados por los algoritmos para determinar los diferentes grupos de casos que tienen características comunes. La creación de estos grupos permite la extracción de información desde los conjuntos de datos disponibles e identifica algunas características que la información oculta [26].

2.8.1.3. Aprendizaje semisupervisado

En este caso el conjunto de datos para entrenamiento contiene una gran cantidad de ejemplos sin etiquetar y un pequeño número de ejemplos etiquetados [53].

2.8.2. Aprendizaje profundo - Deep Learning (DL)

El Deep Learning (DL) es un subconjunto del ML que se basa en las ANN que fueron diseñadas inspirándose en la estructura y funcionamiento del cerebro humano. Este tipo de técnicas es ampliamente usada en campos muy diversos para resolver tareas muy complejas [77, 101].

Esta tecnología se ha caracterizado porque ha evolucionado muy rápidamente, nuevas arquitecturas aparecen cada pocos meses. En la sección 2.9 se analizarán aquellas arquitecturas relevantes para esta investigación [81].

2.8.3. Arquitecturas de aprendizaje profundo

En la presente sección se resumen varias arquitecturas de DL utilizadas que tiene relación con la resolución de problemas de CV y de análisis de secuencias de eventos en el tiempo [62].

Se presenta una Tabla 2.8.3, con un resumen del uso de estas arquitectura según [62]:

Tabla 2.3: Usos de los tipos de redes

Tipo de Red	Usos	Referencias
RNN	Reconocimiento de voz Reconocimiento de escritura Compresión de texto en lenguaje natural Reconocimiento de escritura a mano	M. Tim Jones [62]
LSTM/GRU	Reconocimiento de voz Reconocimiento de gestos Subtítulos de imágenes Reconocimiento de imágenes Análisis de video	M. Tim Jones [62]
CNN	Procesamiento de lenguaje natural Reconocimiento de imágenes Recuperación de información Comprensión del lenguaje natural	M. Tim Jones [62]
DBN	Predicción de fallas Recuperación de información	M. Tim Jones [62]
DSN	Reconocimiento de voz continuo	M. Tim Jones [62]

Se presenta a continuación un breve resumen de cada una de los principales tipos de redes neuronales profundas:

2.8.3.1. Redes neuronales convolucionales (CNN)

Este tipo de arquitectura tuvo su origen en la corteza visual del cerebro de un gato que contiene una compleja secuencia de células. Su aplicación es extensiva para aplicaciones como el procesamiento de imágenes, del lenguaje natural y del habla, entre otras[81].

Redes neuronales convolucionales (CNN) tiene tres ventajas principales:

- Compartir parámetros
- Interacciones dispersas

- Representaciones descriptivas

CNN utiliza un esquema jerárquico para aprender características directamente desde los datos [22]. El entrenamiento requiere de grandes cantidades de datos bien anotados disponibles [40].

CNN tiene dos principales operaciones: convolución y agrupamiento. La convolución es una multiplicación de elementos por una matriz o filtro, para generar un mapa de características extraídas. El agrupamiento es un muestreo descendente para reducir la dimensionalidad, reduciendo el tamaño de las representaciones y conservando la invariancia espacial [22].

2.8.4. Redes neuronales recurrentes (RNN)

La arquitectura de las Redes neuronales recurrentes (RNN) es muy similar a una red "feedforward", excepto que cada unidad oculta calcula una función diferente. En cada momento, la unidad calcula la función de la entrada actual y la de su estado anterior, llamada normalmente "célula estado" [22].

Para utilizar la estructura bidimensional de los datos de entrada (por ejemplo, una imagen), se utilizan conexiones locales y pesos compartidos. A diferencia de las redes completamente conectadas. Este proceso genera muy pocos parámetros, lo que produce que este tipo de red sea más rápida y fácil de entrenar. Esta operación es similar a la de las células de la corteza visual [81]. Este tipo de red, ha demostrado que posee un alto rendimiento en varios dominios al aprender características automáticamente de datos crudos[112].

2.8.4.1. Redes neuronales recursivas (RvNN)

La Redes neuronales recursivas (RvNN) son arquitecturas creadas para procesar objetos que fueron estructurados de una forma arbitraria, tales como arboles o grafos. Permite hacer predicciones en una estructura jerárquica, así como clasificar las salidas utilizando vectores composicionales [81].

2.8.4.2. Redes neuronales Long Short-Term Memory (LSTM)

Las redes Long Short-Term Memory (LSTM) codifican información histórica en unidades de memoria reguladas con puertas no lineales para descubrir dependencias temporales.

Las LSTM son capaces de explotar información temporal de una secuencia de datos con longitud arbitraria mediante el mapeo recursivo de la secuencia de entrada para generar salidas etiquetadas con unidades ocultas. Cada unidad mantiene una celda de memoria, que almacena información a lo largo del tiempo protegida por varias unidades no lineales para controlar la cantidad de cambios e influenciar el contenido de la memoria [112].

Este tipo de red funciona bien para hacer predicciones basadas de series temporales, evitando el problema de dependencia a largo plazo que tienen las RNN tradicionales. LSTM también es adecuada para tareas de clasificación y procesamiento. Se utiliza por ejemplo en aplicaciones como: Google Translate, Apple Siri y Amazon Alexa [111].

2.9. Sistemas de reconocimiento y Localización de objetos

En este apartado se hace un resumen de las principales arquitecturas de DL cuyo fin es la identificación de objetos de tiempo real y por sus características pueden ser incluidas en el módulo de detección de objeto y acciones.

2.9.1. Faster R-CNN (FrRCNN)

Faster R-CNN (FrR-CNN) es la tercera generación de la familia R-CNN, se basa en Fast R-CNN [95, 116]. En comparación con la primera y segunda generación de modelos de la familia R-CNN, el FrR-CNN propone una alternativa basada en CNN, la Red de Propuesta de Región (RPN). Donde su principal característica es que comparte los pesos y sesgos con la red de detección basada en CNN. Esta integración inmediata podría garantizar su capacidad de detección de objetos en tiempo real con gracias a su velocidad y precisión en la detección [95]. Esto se lográ al generar propues-

tas de región, al compartir capas entrecruzadas en lugar de usar Búsqueda Selectiva, lo que reduce la sobrecarga computacional. Sin embargo, estos métodos aún tienen altos costos computacionales debido a la existencia de extracción de características y generación de recomendaciones de región, lo que reducirá la velocidad de inferencia [116].

El enfoque R-CNN más rápido es la tercera generación de la familia R-CNN FrR-CNN, está compuesto por dos módulos.

- Una red profunda totalmente convolucional que propone regiones [89].
- Un detector Fast R-CNN que utiliza las regiones propuestas [89].

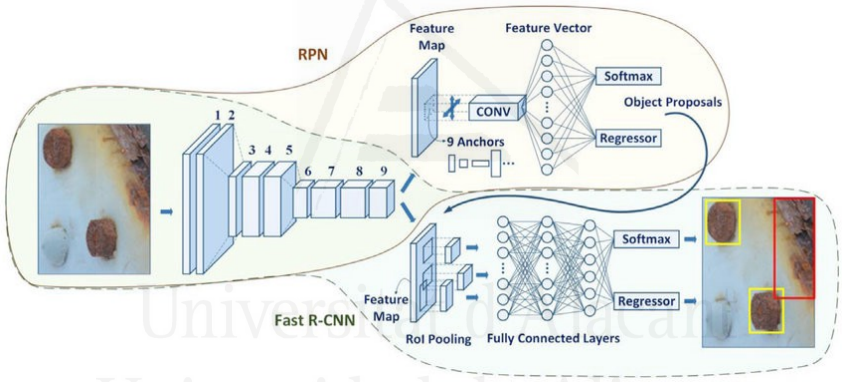


Figura 2.3: Arquitectura de Faster R-CNN (Fuente towardsdatascience.com)

Se inicia el proceso entrenando filtros para extraer las características apropiadas de la imagen. Luego, se aplican las redes de convolución, compuestas por capas de convolución, capas de agrupación, finalizando con una capa completamente conectada u otro elemento que se utilizará para una tarea particular como clasificación o detección. RPN es una pequeña red neuronal que se desliza sobre el último mapa de características de las capas de convolución y predice si hay un objeto o no y también predice el cuadro delimitador de esos objetos. Se finaliza utilizando otras redes neuronales totalmente conectadas que toman como entrada las regiones propuestas por el RPN y predicen la clase de objetos (clasificación) y los cuadros de límites (regresión) [44].

2.9.2. Single Shot MultiBox Detector (SSD)

Este es un método de detección de tipo "single-shot"para múltiples categorías, donde el núcleo de modelo es predecir las ponderaciones de las categorías y los cuadros de desplazamiento para los grupos por defecto de cuadros delimitadores usando pequeños filtros convolucionales para aplicarlos a los mapas de características. Este método mejora la precisión de la detección, además de mejorar la velocidad [31, 58].

La alta velocidad y precisión de Single Shot MultiBox Detector (SSD) con imágenes de resolución relativamente baja se atribuye a las siguientes razones [43]:

- Elimina propuestas de cajas delimitadoras como las que se usan en RCNN
- Incluye un filtro convolucional decreciente progresivo para predecir categorías de objetos y compensaciones en ubicaciones de cuadro delimitador.

La alta precisión de detección en SSD se logra mediante el uso de múltiples cajas o filtros con diferentes tamaños y relación de aspecto para la detección de objetos [43].

El SSD tiene dos componentes:

1. Un modelo backbone

   El modelo de backbone generalmente es una red de clasificación de imágenes, previamente entrenada como extractor de características. Esta es típicamente una red como ResNet entrenada en ImageNet de la cual se ha eliminado la capa final de clasificación totalmente conectada. El backbone da como resultado 256 mapas de características 7x7 para una imagen de entrada [20].

2. Un cabezal SSD

   El cabezal SSD es solo una o más capas convolucionales agregadas al backbone y las salidas se interpretan como las cajas delimitadoras y las clases de objetos en la ubicación espacial de las activaciones de las capas finales.

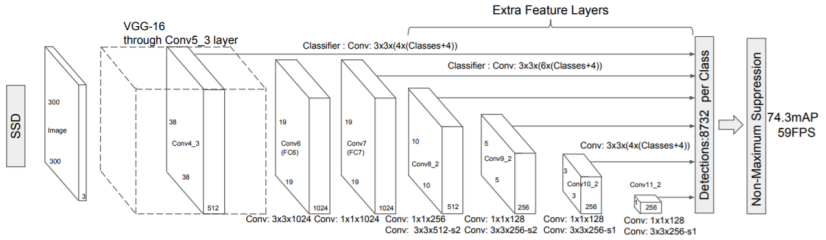


Figura 2.4: Arquitectura de SSD

SSD aprovecha las ventajas de Faster-R-CNN (ver sección 2.9.1), YOLO y pirámides multiescala. Este discretiza el espacio de salida de las cajas delimitadoras en un conjunto de cajas predeterminadas, cada mapa de características tiene sus propias relaciones de aspecto y escalas. Además, este modelo combina predicciones para múltiples mapas de características con diferentes resoluciones para manejar de forma natural objetos de varios tamaños. [102]

2.9.3. You Only Look Once (YOLO)

You Only Look Once (YOLO) es una arquitectura que busca equipar las capacidades que tiene el ser humano en la detección rápida y precisa, que permitiría realizar tareas complejas en tiempo real. Esto permite a los sistemas computacionales realizar acciones como conducción de vehículos sin sensores especializados o vehículos para personas con discapacidad [87]. Una de las principales características de esta arquitectura es permitir la detección y rastreo de múltiple objetos en tiempo real, generando coordenadas para cada objeto. Para lograr esta detección de alta velocidad, se reduce la precisión, aunque el método conserva niveles muy altos de exactitud [3, 60].

Las arquitecturas de tipo YOLO es una estructura típica de red end-to-end. Este tipo de estructura de red es más concisa comparada con las redes de dos etapas de las estructuras tipo R-CNN, que primero generan regiones candidatas y estas realizan la detección y resolución. Integra los mecanismos de detección del área candidata haciendo que la red sea más rápida que sus contrapartes de las arquitecturas de tipo R-CNN [116].

La red que forma la columna vertebral de YOLO es basado en Darknet-53 para extraer características de las imágenes. Toda la red utiliza capas residuales como componentes básicos. Un total de cinco capas residuales con diferentes escalas y pesos, que solamente se ejecutan entre las capas residuales y las de salida [55]. La capa convolucional utiliza núcleos convolucionales alternos de 1 x 1 y 3 x 3 para extraer más características abstractas[11].

El concepto de cuadro de anclaje fue introducido por "Faster RFCNN" (ver sección 2.9.1) y k-means, el cual es usado por YOLO v3 para determinar el tamaño del radio para el cuadro de anclaje que localiza el objeto buscado. En lugar de mapear directamente las coordinadas en el cuadro delimitador, los parámetros son relativos al cuadro de anclaje que fue predicho [55].

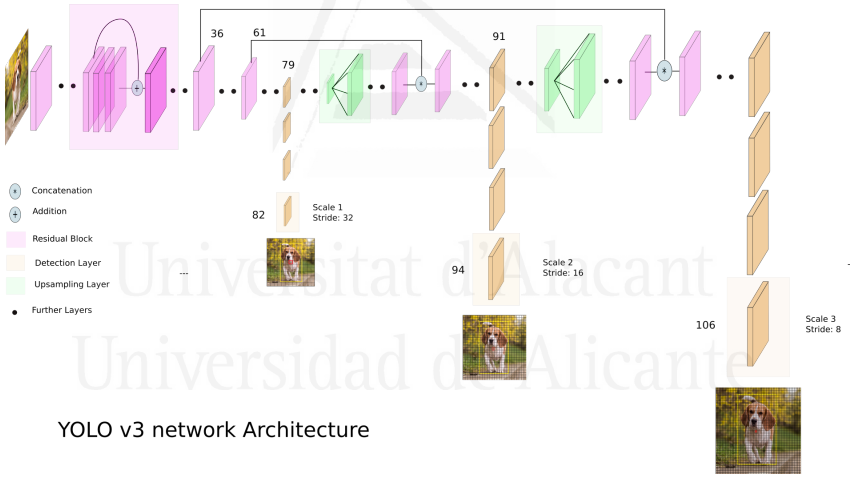


Figura 2.5: Arquitectura de YOLO v3 (Fuente towardsdatascience.com)

2.10. Reconocimiento de acciones del operario

El reconocimiento de acciones es un elemento crucial para este trabajo de investigación, debido a que uno de los puntos relevantes es lograr identificar qué acciones está realizando un operario. De esta forma, será posible determinar si lo hace de la forma según las especificaciones del instructivo de como manufacturar un producto.

Según Yang et al. [119], se tienen dos áreas básicas dentro de la CV las cuales son: el reconocimiento y la predicción de acciones. La primera es capaz de reconocer las acciones llevadas a cabo por un sujeto en un vídeo que contiene la ejecución completa de la acción. La segunda es capaz de predecir acciones a partir de datos de vídeo temporalmente incompletos. Tiene una amplía diversidad de usos en las áreas de la videovigilancia, la vida asistida por el entorno, espacios de compras inteligentes, etc. Abellan-Abenza et al. [2].

Dependiendo del tipo de entrada visual, es posible dividirlo en imágenes 2D y 3D. Actualmente, dado que las imágenes 2D no son las adecuadas para reconocer todos los tipos de acciones, se propone utilizar cámaras 3D (RGB-D, LIDAR, etc.) para mejorar el reconocimiento de acciones [79].

Algunos investigadores comoYan et al. [117], están utilizando el reconocimiento de acciones, para determinar elementos relevantes en el estudio del método de trabajo, tales como: postura de trabajo, atención laboral y fatiga mental. Aunque en esta investigación se enfoca a la identificación del seguimiento de las acciones dentro de un proceso de ensamble, no se consideran los elementos que podrían afectar el rendimiento en las labores.

Investigadores como Yang et al. [118], explican para la detección de acciones en humanos consistente principalmente en dos elementos. El primero elemento es la extracción de la característica de acción, que normalmente utiliza características geométricas humanas. Y el segundo elemento, es la detección de movimiento que busca identificar los límites para obtener el esqueleto estrella con dos indicadores, la posición del cuerpo y el movimiento circular de los segmentos del cuerpo. En actualidad la mayor parte de estos análisis se realizan en entornos de dos dimensiones, pero gracias a la mejora en las técnicas y el hardware autores como Park and Kim [79] proponen información en 3 dimensiones para mejorar el reconocimiento de las acciones. En cambio autores como Abellan-Abenza et al. [2] que al utilizar distintas combinaciones de características de movimiento, características profundas y características estáticas se puede obtener mejoras significativas en el reconocimiento de acciones. Kong and Fu [45] proponen fusionar múltiples tipos de funciones de vídeos es una forma efectiva para el reconocimiento de acciones.

2.11. Técnicas tipo "Image Captioning" para etiquetado de vídeos

Image Captioning es una técnica que busca generar una descripción textual automática de una imagen [59]. Dicha descripción se pueden utilizar para describir lo que pasa en entornos de ensamble, por ejemplo verificando los pasos necesarios en un proceso de manufactura.

Pero en determinadas circunstancias se quiere más que la descripción con una simple etiqueta, es decir, se requiere la descripción completa de acciones. Por ello investigadores como Wang. et al. [109], han desarrollado trabajos en captioning para vídeo, donde utilizan técnicas de Hierarchical Reinforcement Learning para la generación de descripciones.

Por su parte, Krishna et al. [46], están utilizando técnicas de long short-term memory (LSTM) en un modelo de dense-captioning de detección de eventos.

En ambos casos se usan narrativas de descripción en lenguaje natural. Yao et al. [120], propone crear las narrativas en diferentes formas: métodos basados en plantillas, donde se crean estructuras y se busca generar narrativas en formas fijas. Se emplean enfoques por búsqueda de elementos visuales similares etiquetados y se copian estos textos y modelos basados en lenguajes que utilizan modelos de recuperación tipo k-nearest neighbor.

Para que resulte útil, se requiere contar con una técnica que permita dar sentido a las palabras usadas en la descripción. Para esto resulta conveniente contar con sistemas gramaticales para que estructure las instrucciones de tal forma que se logren expresar reglas aplicables a la industria.

En el mundo de la robótica, hacer que un robot realice las mismas acciones que un operario humano, solo observándolo, es un reto muy avanzado. Autores como Nguyen et al. [73], están realizando propuestas para entender e imitar las acciones humanas, sin definir objetivos o validaciones en las acciones.

Dadas las investigaciones anteriores, se identifica la necesidad de contar con una gramática que permita estructurar instrucciones y que facilite la comunicación adecuada de las acciones que se quieren realizar, sin generar ambigüedades entre la partes involucradas. Para mitigar estas deficiencias,

Yang et Al. [119], proponen un sistema de redes neuronales convolucionales, que a través del análisis de vídeos construyen arboles de gramática de las acciones observadas. Estas son gramáticas de uso general no restringido para un uso particular, por lo que podría generar acciones no estructuradas para la verificación estricta de lo captado.

Investigadores como Mancini et al. [64], están realizando trabajos en la detección de objetos en dominios específicos de industria, pero sin la definición de una gramática para describir la secuencia de acciones. Por lo que la propuesta de crear una gramática de dominio específico de ensambles en Industria 4.0 resulta una idea novedosa.

Capítulo 3

Lenguaje de Descripción de procesos de manufactura

En este capítulo se propone el diseño de un lenguaje que permite describir las tareas que realiza un operario en un área de trabajo para procesos de manufactura. Este lenguaje fue concebido para que fuera de uso general en la manufactura y pueda ser extendido, agregando nuevas acciones, herramientas, componentes y piezas. Además, este lenguaje puede ser configurable en distintos niveles. Entre los posibles elementos configurables se considera: si los operarios son diestros o zurdos, la localización de elementos en el área de trabajo, especificada manualmente, o automáticamente mediante el sistema de control visual que se detalla en el Capítulo 5. Se expone también la descripción de la gramática propuesta para el lenguaje, ejemplos de un proceso real llevado al lenguaje propuesto y validado empíricamente. Finalmente se hace un resumen de las características propias del lenguaje, así como un ejemplo extendido de su uso.

3.1. Introducción

Esta sección se centra en la propuesta de la creación de un lenguaje que permite estandarizar la forma de describir las acciones que realiza el operario durante los procesos de ensamblaje manual. Con el fin de lograr homogeneidad en los procesos de ensamblaje de los productos.

En las fábricas, el ensamblaje de productos o componentes por parte de los operadores es una tarea compleja que no está exenta de problemas recurrentes. En este proceso, los operadores a menudo cometen errores que pueden conducir a productos defectuosos. Esto provoca que se deben inspeccionar más tarde para verificar su correcto ensamblaje y complican el control de calidad.

Los principales problemas son causados por varias razones, incluida la alta rotación de empleados debido a la falta de experiencia en la fabricación de productos específicos o la manera confusa o ambigua de documentar las instrucciones para componentes similares. El fin del lenguaje propuesto es permitir minimizar las pérdidas por defectos de fabricación y el aumento de tiempo y de dinero empleados por descartes y repeticiones en los ensambles.

Este lenguaje puede ser utilizado de forma independiente al resto de aportes de esta investigación, ya que, los operarios en una industria pueden utilizarlo para documentar sus procesos. Pero el principal aporte de este lenguaje para la investigación doctoral es que sirva como uno de las entradas al sistema para el control visual de ensambles. En concreto se plantea como el patrón contra el cual se evaluarán las acciones desarrolladas por el operario para poder determinar si esta realizando las actividades del proceso conforme las especificaciones. Lo cual va permitir minimizar la incidencia de los errores durante el montaje.

Como se mencionó en la sección 2.6.2, una de las técnicas comúnmente utilizadas para el control de calidad son los sistemas de inspección automática. Donde normalmente se comparan los ensambles contra estándares de medición para las partes, pero no logran evaluar si el proceso completo fue realizado correctamente.

Entre las características propias de los entornos de manufactura en la Industria 4.0, como se pueden apreciar en la sección 2.6, se menciona

que muchos de los productos que se fabrican son personalizados y de alta calidad, por lo que, el hecho de poder definir un control de ensambles se puede convertir en una ventaja competitiva para las Micro, Pequeñas y Medianas Empresas (MiPyME).

También se identificó en el Capítulo 2, que en la inspección automática, se aplica la CV, la cual permite hacer un uso extensivo de técnicas de DL para el procesamiento visual [35].

3.2. Teoría de Micromovimientos Therbligs

En este punto se presenta un breve resumen de la teoría de micromovimientos Therbligs, que conforma uno de los fundamentos de la II, que también sirve como base en la propuesta de este nuevo lenguaje estructurado de descripción de acciones de ensambles en manufactura.

Para plantear el lenguaje, se investigó la manera habitual de registro de las instrucciones o acciones manuales en la industria. Se identificó que los esquemas gráficos son una herramienta habituales para realizar esta tarea, por ejemplo los distintos tipos de flujogramas. Sin embargo, para realizar un lenguaje formal y que fuera más general, se determinó que se requería una definición más detallada de los movimientos manuales. Se tomó como base la Teoría de registro de micro-movimientos Therbligs de Frank y Lillian Gilbreth, que fue desarrollada entre 1908 y 1924, donde se proponen 18 acciones básicas [33][105][24].

3.3. Lenguaje de Descripción de Manufactura)

En esta sección se detalla la propuesta Lenguaje de Descripción de Manufactura - Manufacturing Description Language (MDL), un lenguaje que permite a los encargados de las fábricas documentar las secuencias de actividades necesarias para el desarrollo de un producto.

Figura 3.1: Acciones de Therblig (Fuente wikipedia)

De acuerdo con RAE, se define lenguaje como: "Facultad del ser humano de expresarse y comunicarse con los demás a través del sonido articulado o de otros sistemas de signos" y "Conjunto de signos y reglas que permite la comunicación con una computadora" [1]. Con esta definición se introducen conceptos relacionados para el resto de la investigación.

Se requiere del uso de lenguajes para poder expresar las instrucciones a través de una forma correcta según las ideas que se buscan transmitir, para esto es requerido contar con una semántica particular propia de cada lenguaje. Por lo que se hizo una revisión de la literatura para buscar si existían lenguajes particulares que lograran representar las instrucciones necesarias para el ensamble de productos por parte de los operarios.

Dentro de los hallazgos más significativos se encontró una propuesta de IBM a mediados de los años 80 se hizo la propuesta de un lenguaje para el proceso de ensamble de productos, pero restringidas al contexto de aquel tiempo, ya que se asumía se eran solo los robots que realizarían los trabajos, operarios no realizarían este trabajo, los robots eran máquinas tipo CNC que realizan trabajos repetitivos de posiciones exactas, sin ningún tipo de análisis cognitivo[72].

[1] https://dle.rae.es/lenguaje

En revisión bibliográfica se identificaron investigaciones donde a través de la CV se realiza un proceso de identificación las acciones que generan texto, en algunos casos utilizando Procesamiento del Lenguaje Natural; pero al no tener semántica no genera un formato estandarizado que complicaría el proceso de análisis de acciones orientadas a la manufactura. Algunas de estas investigaciones se mencionan a continuación.

Uno de los retos más avanzados en el DL es hacer que una máquina comprenda las acciones de un humano (en este caso operario) a través de CV, para que una máquina realice las mismas acciones que un operario humano a través del proceso de observación[73]. Una de las formas para hacer que DL puedan reconocer las acciones es a través de la detección de tareas, que puede generar como salida captioning en sentencias utilizando lenguaje natural, pero sin gramática por que esta forma de lenguaje le falta la semántica que no permite expresar correctamente las ideas[73].

Yang et al. [119] están trabajando en sistemas que utilizan representaciones probabilistas de manipulación de gramáticas basadas modulos que pasean las instrucciones con el fin de representar oraciones de forma visual para manipular robots. Para mitigar estas deficiencias, Yang et Al., proponen un sistema de redes neuronales convolucionales, que a través del análisis de vídeos construyen arboles de gramática de las acciones observadas[119]. Estas son gramáticas de uso general no restringido para un uso particular, por lo que podría generar acciones no estructuradas para la verificación estricta de lo captado. Ya investigadores como Mancini et al., están realizando trabajos en la detección de objetos en dominios específicos de industria [64], pero sin la definición de una gramática para describir la secuencia de acciones. Por lo que la propuesta de crear una gramática de dominio específico de ensambles en Industria 4.0 resulta una idea novedosa.

En desarrollo de trabajos para la colaboración de humanos-robot, resulta relevante que le sea natural a la personas.

3.4. Detalle del Lenguaje de Descripción de Manufactura - Manufacturing Description Language (MDL)

El Lenguaje de Descripción de Manufactura - Manufacturing Description Language (MDL), es un lenguaje que fue diseñado como complemento para esta investigación y además que pueda ser de uso general para descripción procesos de manufactura.

Durante el diseño del lenguaje se estableció que formará parte básica de una solución global, que utilizaría sistemas de IA, para indicar las configuraciones generales de operación de forma automática, como por ejemplo: las restricciones del área de trabajo, herramientas a usar, piezas y componentes necesarios para el ensamble. Pero para los entornos donde no se tenga estas capacidades el lenguaje provee una sección de setup, donde se pueden establecer los parámetros de forma manual.

Otra consideración del lenguaje durante el diseño, fue la estructura sintáctica de las instrucciones, que tenían que sencillas, concisas, que requieran un breve entrenamiento para su uso, en especial porque puede ser usado por personas de distintas especialidades, como los encargados de calidad, producción y los propios operarios de las celdas de producción.

Entre las características relevantes a destacar es que el sistema permite definir ensamble por componentes, esto genera una ventaja ya que se pueden establecer hito durante la construcción, para chequeos de calidad durante el ensamble; también permite establecer rutas alternas de ensamble.

Para lo cual se definió un conjunto de básico de herramientas, pero el sistema da posibilidad de extender con nuevas configuraciones, entre las que se incorporan dentro de la versión básica del lenguaje se tienen (sus nombres son en inglés para motivar su utilización universal):

- Clamp
- Gun Drill
- Screw Drill
- Ball Pein Hammer
- Claw Hammer
- Nut Driver
- Diagonal Pliers
- Lineman Pliers

- Locking Pliers
- Long Nose Pliers
- Ratchet
- Electric Screwdriver
- Phillips Screwdriver
- Slotted Screwdriver
- Socket
- Adjustable Wrench
- Allen Wrench
- Combination Wrench

Además, como complemento de las herramientas, algunas cuentan con accesorios para los cuales fueron definidas en el lenguaje, así como partes y elementos de uso general, entre estos se encuentran:

- Bolt
- Bit Drill
- Gears
- Nut
- Screw
- Washer

Utilizando la teoría expresada en la sección 3.2 y el estudio de la forma en que las empresas documentan sus procesos, se propone una gramática sencilla que permita describir las actividades cotidianas en el ensamble de productos, teniendo en consideración las necesidades de personas que trabajan en está área.

Para representar secuencias completas de ensamble se utilizan las instrucciones desarrolladas con las primitivas de micro-movimientos. Con el fin de incentivar su uso en los profesionales de la II y las ventajas de usar técnicas comprobadas.

Además, dado que este lenguaje se planteó como base de un sistema de control visual, algunas de sus características y estructura fueron pensadas para que tuviera una gramática formal y pudiera ser analizado por un interprete que permita compararlo con las entradas visuales del sistema de control.

Por tanto, el principal objetivo del lenguaje es evaluar si un operario esta ensamblando un componente o producto según las especificaciones dadas en las instrucciones.

Esta comprobación se realiza en dos etapas:

1. Etapa 1, consiste en que un experto, por ejemplo un ingeniero en procesos o de calidad, utilice el lenguaje para describir las acciones

necesarias para el ensamble de un producto.

2. Etapa 2, el operario desarrolla sus labores de ensamblaje, mientras un sistema de visión artificial registra como entrada lo que esta ocurriendo en el área de trabajo, lo procesa y convierte en una descripción textual mediante técnicas de captioning de vídeo.

Posteriormente se realiza una comparación con la descripción del lenguaje, para determinar si el operario sigue las instrucciones.

Además, el sistema es capaz de determinar el paso o acción actual, con el fin de indicar al operario los pasos siguientes y disminuir la posibilidad de errores al realizar el ensamble. La propuesta completa del lenguaje se puede encontrar en GitHub. `https://github.com/mazamorahdez/manufacturing_language`.

3.4.1. Otros conjuntos de datos relevantes de uso específico

Investigadores como Starke et al. [99], están desarrollando trabajos muy interesantes que se pueden explorar en futuras líneas de investigación como continuación a este trabajo ,ya que estos investigadores hacen estudios dentro del área de factores humanos, generando dataset de la posiciones de las manos, para la ejecución de acciones sobre ensambles y herramientas a nivel de presión de agarre y movimiento de los esqueletos, que podría mejorar la definición de acciones de manufactura y su optimización en los procesos de ensamble.

En este apartado se hará una descripción general de las secciones que componen el lenguaje: en la primera sección se explican las instrucciones que permiten describir elementos propios del entorno de trabajo y posteriormente se describe la sección que incluye aquellas instrucciones que permite describir las actividades del proceso de ensamblaje manual.

3.4.2. Parámetros de operación

En esta sección se detalla el modo en que se realizan las configuraciones iniciales de ejecución dentro del lenguaje, en dos modos distintos de operación: automático y manual.

En el diseño del lenguaje, se consideraron dos modos de operación, en el primero donde se auto-configurará el sistema a través de CV, que permite que el sistema describa el entorno inicial de trabajo, es decir, a través de la CV se identifican los elementos presentes y sus ubicaciones. En el último de los escenarios, se requiere que un operario o ingeniero de procesos describa el área de trabajo utilizando los elementos que el lenguaje le provee, que se detallan a continuación.

Para realizar la configuración manual se implementan las instrucciones que se pueden apreciar detalladamente en (código 3.1):

1. *product*, establece el código o nombre del producto que se desea registrar en las instrucciones de ensamble.

2. *setup*, se colocan las posiciones iniciales dentro del área de trabajo de las partes, componentes y herramientas. Además, se indica la ubicación de los componentes (ensambles y sub-ensambles) existentes; así como la cantidad de componentes que se van a generar durante el proceso de manufactura. En esta misma sección del código, se define la mano dominante para el operario, de modo que el sistema configurará las instrucciones según las características individuales de cada usuario.

Listing 3.1: Parámetros de operación

```
1  <set> := assembly #<identifier> [to-create] [in <coordinate>:<offset>:<unit>];
2  <set> := hand <x-position>;
3  <set> := bin <part> [in <coordinate>:<offset>:<unit>];
4  <set> := tool <tool> [in <coordinate>:<offset>:<unit>];
5  <set> := accessory <accessory> [in <coordinate>:<offset>:<unit>];
```

En la gramática del lenguaje se puede encontrar la sección “Set up”, esta sección se inicia con la instrucción setup-begin <sets>setup-end. Donde los “sets” pueden tomar alguna de las siguientes opciones:

- *assembly*, establece la ubicación de un ensamble que se usará durante el proceso de manufactura; en caso que se indique to_create significa que el ensamble se creará a partir de la unión de 2 o más ensambles durante la ejecución. De modo que en el sistema se definan los bloques correspondientes para estos ensambles.

- *hand*, para indicar al sistema cual es la mano dominante del operario, ajustando así las instrucciones de mano dominante y no-dominante según cada individuo.
- *bin*, define la ubicación del contenedor y el contenido, para que cuando se establezcan las instrucciones el sistema conozca de donde debe tomar o en donde colocar los insumos del ensamble.
- *accessory*, algunas herramientas utilizan accesorios. Así se le indica la posición de estos al sistema, para que cuando se indique en las instrucciones pueda verificar si se utilizaron los accesorios correctos.

3.4.3. Acciones de ensamble

En esta sección se introducen las instrucciones que permiten realizar la descripción del proceso de ensamblaje, a cada instrucción dentro de este segmento del código se le denomina "step".

Cada instrucción de tipo "step", está a su vez clasificada en dos posible tipos. Aquellas instrucciones que definen una operación unitaria, que llamaremos "Pasos individuales". Y a las instrucciones que definen conjuntos de pasos les llamaremos "Bloques de pasos". Estos se detallan en los puntos siguientes.

3.4.3.1. Pasos individuales

En este punto se detallan, aquellas instrucciones que permiten realizar la definición de una acción individual. Dentro de este tipo de acciones se tienen tres categorías, que se definen a continuación:

3.4.3.1.1. hand , estas operaciones tienen relación con los movimientos o acciones realizadas con las manos, se puede ver en el código 3.2.

Listing 3.2: Pasos manuales

```
1 <step> := <hand-action>:(<part>|<tool>) with <handused>
2 [in assembly #<identifier>];
3 <step> := <hand-action>:<handused> [in <coordinate>:<offset>:<unit>];
4 <step> := <hand-action>:assembly #<identifier> with
5  (assembly #<identifier> | <handused>);
6 <step> := move:assembly #<identifier> with <handused> from <coordinate>:<offset>
7 to <coordinate>:<offset>:<unit>;
```

```
8  <handused> := hand | hand-nondominant | hand-any | hand-both
9  <hand-action> := put | hold | take | grip | release | push | spin | turn | join | move
```

3.4.3.1.2. tool , estas instrucciones tienen relación con el uso de las herramientas o sus accesorios, como se puede apreciar en el código 3.3.

Listing 3.3: Operaciones con herramientas.

```
1  <step> := <substep> [in <coordinate>:<offset>:<unit>];
2  <substep> := <hammer-action>:<hammer> | <wrench-action>:<wrench>
3  | <screwdriver-action>:<screwdriver> | <pliers-action>:<pliers>
4  | <driller-action>:<driller> with <accessory> | <clamp-action>:clamp
5  | <ratchet-action>:ratchet with (socket|none) | <screwdriver-action>:nut_driver
```

3.4.3.1.3. move , estas operaciones tienen como fin definir desplazamientos de los ensamble en el área de trabajo. Las instrucciones para esta tarea se pueden observar en el código 3.4.

Listing 3.4: Operaciones movimientos.

```
1  <step> := move:assembly #<identifier> with <handused> from <coordinate>:<offset>
2  to <coordinate>:<offset>:<unit>;
```


3.4.3.2. Bloques de pasos

Dentro del lenguaje se definen agrupaciones de instrucciones que trabajan en conjunto para realizar un tarea particular durante los ensambles, se puede observar las instrucciones en el código 3.5.

Estas secuencias se componen de cualquier instrucción que sea de tipo acción de ensamble, como las detalladas en la sección 3.4.3. Entre los bloques disponibles se tiene los siguientes tipos:

3.4.3.2.1. make-assembly , este es uno de los bloques más relevantes del lenguaje, ya que permite describir el concepto de ensamble (o subensamble), que definen hitos durante el proceso. Define la construcción de elementos complejos a partir de la unión de otros más básicos. Un elemento relevante de este bloque es que define unidades con nombre, que se pueden invocar en otros "assembly".

3.4.3.2.2. repetition , existen conjuntos de pasos (incluyendo bloques), que se repiten varias veces. En estos casos se define el conjunto de pasos a repetir y la cantidad de veces que se deben ejecutar.

3.4.3.2.3. any-order , una premisa básica del lenguaje es que las sentencias se ejecuta en el orden descrito, a menos que se indique lo contrario. Para ello se usa el bloque any-order que le indica al sistema que se pueden ejecutar las instrucciones en cualquier orden.

3.4.3.2.4. parallel , es usada cuando se necesita ejecutar instrucciones de forma paralela: ya sea porque existen más de un operario interactuando en el área de trabajo, o para describir que varias acciones pueden ser ejecutadas por ambas manos al mismo tiempo.

Listing 3.5: Bloques de ejecución

```
1  <make-assembly> := assembly-start #<identifier>:<steps> assembly-end;
2  <repetition> := repeat:<steps> until <digits> times;
3  <in-any-order> := any-order-begin <steps> any-order-end;
4  <parallel> := parallel-begin: <steps> parallel-end;
```

Listing 3.6: Símbolos del lenguaje

```
1  <offset> := <digits>
2  <unit> := mm | cm | mm
3  <coordinate> := <sign><digits>,<sign><digits>
4  <sign> := <void> | <positive> | <negative>
5  <void> := ''
6  <positive> := +
7  <negative> := -
8  <position> := <x-position> | <y-position> | <y-position>-<x-position>
9  <x-position> := rigth | left
10 <y-position> := upper | lower
11 <identifier> := <char> | <char><word> | #bytes#
12 <word> := <alpha><word>
13 <alpha> := <char> | <digit> | <void>
14 <char> := a | b | ... | z | A | B |...| Z | _ | - | & | ' ' | . | , | @
15 <digits> := <digit> | <digit><digits>
16 <digit> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Otros elementos cruciales que intervienen en la definición del lenguaje, y que son de uso general en la parte de parametrización y código ejecutable, son los que se describen a continuación:

- hand-action: estas son la definición de todas las acciones posibles a realizar con las manos. Algunas de estas fueron tomadas de los Therbligs, otras fueron incorporadas acorde a la realidad actual.

- tool: esta define la lista de herramientas con las que se pueden realizar transformaciones o trabajos sobre los elementos en el ensamble de productos. Se crearon familias básicas y comunes que se determinaron con un estudio a operarios de ensambles manuales en industrias varias. Sin embargo, el lenguaje esta diseñado para poder incorporar más herramientas, debido principalmente al desarrollo constante de nuevos equipos. Se puede ver el listado básico de las herramientas en el código 3.7.

- tool-action: son las acciones que se pueden realizar sobre, o mediante, las herramientas. Como no todas las herramientas comparten la misma gama de acciones, existe un elemento que permite realizar esta asociación. Al igual que las herramientas, esta sección se puede actualizar para representar las acciones de las nuevas herramientas que se incorporen al mercado. El sistema tiene la capacidad de extensión en las acciones, pero se define un conjunto básico que se puede apreciar en el código 3.7.

- substep : esta es las sección donde se realiza la unión de las acciones propias de una herramienta con la respectiva familia de herramientas.

- part : las partes son los elementos más simples y comunes que se utilizan en los ensambles, al igual que el resto de los elementos, estos pueden ampliarse y ser modificados. El sistema ya incorpora una lista básica que contiene: tornillos, tuercas, arandelas, entre otros.

- accessory : un accesorio se define como un complemento para una herramienta particular, al igual que tool-actions, son particulares a cada herramienta, por lo que también se debe establecer su relación, en este caso se hace en substep. Algunos ejemplos de accesorios son brocas y cubos de tuercas, entre otros.

- coordinate : permite ubicar un par ordenado para localizar los elementos en la mesa, donde se asume el centroide de la mesa de trabajo como el punto (0,0). Se parte del supuesto de que la cámara de control visual se ubica sobre la mesa de trabajo, lo que hace posible interpretar el espacio de trabajo como un plano.

- offset : para establecer la coordenada, se define el offset, que se establece como la longitud de cada uno de lo lados de un cuadrado, donde el centroide del cuadrado es la coordenada.
- unit : son las unidades con las que trabajará el offset y las coordenadas.

Listing 3.7: Herramientas y sus acciones

```
6  <tool> := <hammer> | <wrench> | <screwdriver>
7  | <pliers> | <driller> | ratchet| clamp
8  | nut_driver
9  <hammer> := hammer_ball_pein | hammer_claw
10 <wrench> := wrench_adjustable | wrench_allen | wrench_combination
11 <screwdriver> := screwdriver_electric |
12 screwdriver_phillips | screwdriver_slotted
13 <pliers> := pliers_diagonal | pliers_lineman | pliers_locking | pliers_long_nose
14 <driller> := drill_gun | drill_screw
15 <hammer-action> := nail | hammer_out | hit
16 <wrench-action> := pull | tight | locknut
17 <screwdriver-action> := screw
18 <pliers-action> := loosen | cut | hold | tighten
19 <driller-action> := drilling
20 <clamp-action> := loosen | tighten
21 <ratchet-action> := turn
```

3.5. Validación de la Propuesta

Para realizar la validación del lenguaje, se diseño un experimento con un grupo de industrias, cuya forma de trabajo fuera a través de celdas de manufactura. Para lo cual cada una de las industrias facilito un representante. A cada representante participó en una entrevista que tenía como objetivo identificar la forma de como representan las instrucciones durante el proceso de ensamble, además de cuales son las herramientas que utilizaban normalmente para llevar a cabo estas labores.

Para el diseño del experimento se establecieron varias fases para realizar la validación, que tenía como objetivo generar un lenguaje que fuera equivalente conceptualmente a las instrucciones regulares de cada fabrica.

La primera fase del experimento se detallan a continuación:

1. Identificar ejemplos de secuencias de producción por industria para generar la comprobación.

   Durante la investigación se recopilo diferentes formas de como las empresas documentan sus procesos y la forma de como ensamblan

sus productos. Se tomaron esta información se procedió a generar los equivalentes de las instrucciones de cada empresa utilizando el Manufacturing Description Language.

2. Para cada secuencias de la empresa se realizó una traducción de las instrucciones al lenguaje creado; para evaluar el nivel de equivalencia entre los dos.

   Con ayuda de contrapartes en las empresas participantes en el experimento, se revisaron las instrucciones generadas con las versiones originales para determinar el nivel de exactitud, se logró determinar que aproximadamente el 97 % de las instrucciones se lograron convertir, el restante 3 % corresponde a elementos muy particulares de cada industria. Para solucionar este punto, se procedió a generar una extensión propia al lenguaje; es importante indicar que esa fue una consideración que se tomo como un requerimiento al momento de crear el lenguaje que fuera extensible con la incorporación de nuevas herramientas y acciones.

3. Se hizo una prueba tomando un grupo de operarios y se les ofrece una breve introducción sobre el uso del lenguaje. Con el fin de ejecutar las instrucciones según el nuevo lenguaje para ejecutarlo en las estaciones de trabajo.

La última fase del experimento, el lenguaje se usa para describir un proceso de ensamble, la segunda fase de verificación del lenguaje se estableció tomando procesos de ensamble completos. El equipo de investigación grabó varios videos y generó el código completo de las instrucciones para reproducir los procesos. Estos códigos se pueden ver en los listados 3.8 y 3.9. Este último muestra un par de ejemplos del código reducido del lenguaje de descripción de fabricación generado a partir de los dos videos utilizados para el primer experimento.

Listing 3.8: Código de muestra del primer video de validación

```
1        assemble-begin
2        example01;
3        setup-begin
4        hand right;
5        assembly #truck in 55:-28:cm;
```

```
6           assembly #wheel1 in 14:60:cm;
7           assembly #wheel2 in 14:45:cm;
8           assembly #base_plate in 0:5:cm;
9           tool hammer_ball_pein;
10          tool clamp;
11          tool pliers_diagonal;
12          tool pliers_long_nose;
13          tool screwdriver_slotted;
14          tool screwdriver_slotted;
15          tool screwdriver_phillips;
16          tool wrench_combination;
17          tool wrench_combination;
18          tool wrench_adjustable;
19          bin bolt in -50:-30:cm;
20          bin washer in -32:-6:cm
21          bin bolt in -30:30:cm;
22          setup-end
23          start
24          // Frame01
25          take:assembly #wheel1 with hand-nondominant;
26          take:assembly #wheel2 with hand;
27          // Frame02
28          move:assembly #truck with hand to 0:5:cm;
29          // Frame03
30          join:assembly #truck with assembly #wheel1;
31          // Frame04
32          hold:assembly #truck with hand-nondominant;
33          hold:wrench_combination with hand;
34          take:washer with hand;
35          // Frame05
36          put:washer with hand in assembly #truck;
37          // Frame06
38          tight:wrench_combination in assembly #truck;
39          // Frame07
40          take:nut with hand;
41          // Frame08
42          put:nut with hand in assembly #truck;
43          tight:wrench_combination in assembly #truck;
44          // Frame09
45          move:assembly #truck with hand to 0:5:cm;
46          // Frame10
47          take:bolt with hand;
48          take:assembly #base_plate with hand-nondominant;
49          // Frame11
50          join:bolt with assembly #base_plate;
51          // Frame12
52          take:washer with hand;
53          // Frame13
54          put:washer with hand in assembly #base_plate;
55          // Frame14
56          hold:pliers_long_nose with hand;
57          // Frame15
58          hold:pliers_long_nose in assembly #base_plate;
59          // Frame16
60          take:assembly #base_plate with hand-both;
61          // Frame17
62          join:assembly #truck with assembly #base_plate;
63          // Frame18
64          put:nut with hand in assembly #truck;
65          // Frame19
66          hold:pliers_long_nose with hand;
67          // Frame20
68          hold:pliers_long_nose in assembly #base_plate;
69          // Frame21
70          move:assembly #base_plate with hand to 0:0:cm;
71          end assemble-end
```

Listing 3.9: Sample code from the second validation video

```
assemble-begin
example02;
setup-begin
hand right;
assembly #truck in 5::cm;
assembly #desk in 0:60:cm;
tool hammer_ball_pein;
tool clamp;
tool pliers_diagonal;
tool pliers_long_nose;
tool screwdriver_slotted;
tool screwdriver_slotted;
tool screwdriver_phillips;
tool wrench_combination;
tool wrench_combination;
tool wrench_adjustable;
bin bolt in -10:-10:cm;
bin nut in -5:-10:cm;
setup-end
start
// Frame01
// Initial frame
// Frame02
take:assembly #truck with hand-nondominant;
take:assembly #desk with hand;
move:assembly #desk with hand to 10:0:cm;
// Frame03
take:nut with hand;
// Frame04
join:assembly #truck with assembly #desk;
// Frame05
take:nut with hand-nondominant;
take:screwdriver_slotted with hand;
// Frame06
put:nut with hand in assembly #desk;
// Frame07
take:wrench_combination with hand-nondominant;
// Frame08
locknut:wrench_combination with hand-nondominant in assembly #truck;
screw:screwdriver_slotted with hand in assembly #desk;
// Frame09
take:nut with hand-nondominant;
take:screwdriver_slotted with hand;
// Frame10
locknut:wrench_combination with hand-nondominant in assembly #truck;
screw:screwdriver_slotted with hand in assembly #desk;
// Frame11
spin:assembly #desk with any-hand;
// Frame12
take:nut with hand-nondominant;
// Frame13
put:nut with hand-nondominant in assembly #desk;
// Frame14
take:bolt with hand-nondominant;
// Frame15
put:nut with hand-nondominant in assembly #desk;
// Frame16
take:screwdriver_slotted with hand-nondominant;
// Frame17
take:nut with hand-nondominant;
// Frame18
put:nut with any-hand in assembly #desk;
// Frame19
take:bolt with hand-nondominant;
```

```
65          // Frame20
66          put:bolt with hand-nondominant in assembly #desk;
67          // Frame21
68          take:screwdriver_slotted with hand;
69          end
70          assemble-end
```

Como resultado de esta validación, se pueden apreciar en las instrucciones generadas con la gramática propuesta en los códigos demostrativos presentados a continuación.

Listing 3.10: Gramática del lenguaje

```
1  <S> := <assemble>
2  <assemble> := assemble-begin <product> <setup> start <steps> end assemble-end
3  <product> := <word>;
4  <setup> := setup-begin <sets> setup-end
5  <sets> := <set> | <set><sets>
6  <set> := assembly #<identifier> [to-create] [in <coordinate>:<offset>:<unit>];
7  <set> := hand <x-position>;
8  <set> := bin <part> [in <coordinate>:<offset>:<unit>];
9  <set> := tool <tool> [in <coordinate>:<offset>:<unit>];
10 <set> := accessory <accessory> [in <coordinate>:<offset>:<unit>];
11 <offset> := <digits>
12 <unit> := mm | cm | mm
13 <steps> := <step> | <step><steps>
14 <step> := <make-assembly>
15 <step> := <hand-action>:(<part>|<tool>) with <handused>
16 [in assembly #<identifier>];
17 <step> := <hand-action>:<handused> [in <coordinate>:<offset>:<unit>]
18 [in assembly #<identifier>];
19 <step> := <hand-action>:assembly #<identifier> with
20  (assembly #<identifier> | <handused>);
21 <step> := move:assembly #<identifier> with <handused> from <coordinate>:<offset>
22 to <coordinate>:<offset>:<unit>;
23 <step> := <substep> [in <coordinate>:<offset>:<unit>]
24 [in assembly #<identifier>];
25 <step> := <repetition>
26 <step> := <any-order>
27 <step> := <parallel>
28 <substep> := <hammer-action>:<hammer> | <wrench-action>:<wrench>
29 | <screwdriver-action>:<screwdriver> | <pliers-action>:<pliers>
30 | <driller-action>:<driller> with <accessory> | <clamp-action>:clamp
31 | <ratchet-action>:ratchet with (socket|none) | <screwdriver-action>:nut_driver
32 <handused> := hand | hand-nondominant | hand-any | hand-both
33 <hand-action> := put | hold | take | grip | release
34 | push | spin | turn | join |move
35 <part> := bolt | gears | nut | screw | washer
36 <accessory> := drill_bit | socket | none
37 <make-assembly> := assembly-start #<identifier>:<steps> assembly-end;
38 <repetition> := repeat:<steps> until <digits> times;
39 <parallel> := parallel-begin: <steps> parallel-end;
40 <in-any-order> := any-order-begin <steps> any-order-end;
41 <tool> := <hammer> | <wrench> | <screwdriver>
42 | <pliers> | <driller> | ratchet| clamp
43 | nut_driver
44 <hammer> := hammer_ball_pein | hammer_claw
45 <wrench> := wrench_adjustable | wrench_allen | wrench_combination
46 <screwdriver> := screwdriver_electric
47 | screwdriver_phillips | screwdriver_slotted
48 <pliers> := pliers_diagonal | pliers_lineman | pliers_locking | pliers_long_nose
49 <driller> := drill_gun | drill_screw
50 <hammer-action> := nail | hammer_out | hit
```

```
51 <wrench-action> := pull | tight | locknut
52 <screwdriver-action> := screw
53 <pliers-action> := loosen | cut | hold | tighten
54 <driller-action> := drilling
55 <clamp-action> := loosen | tighten
56 <ratchet-action> := turn
57 <coordinate> := <sign><digits>,<sign><digits>
58 <sign> := <void> | <positive> | <negative>
59 <void> := ''
60 <positive> := +
61 <negative> := -
62 <position> := <x-position> | <y-position> | <y-position>-<x-position>
63 <x-position> := rigth | left
64 <y-position> := upper | lower
65 <identifier> := <char> | <char><word> | #bytes#
66 <word> := <alpha><word>
67 <alpha> := <char> | <digit> | <void>
68 <char> := a | b | ... | z | A | B |...| Z | _ | - | & | ' ' | . | , | @
69 <digits> := <digit> | <digit><digits>
70 <digit> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

3.5.1. Ejemplo 1

En este ejemplo se muestra un acción básica de golpear un elemento con un martillo.

Figura 3.2: Cuadros para ejemplo 1 de MDL

Listing 3.11: Code example

```
1  assemble-begin
2  skateboard;
3  setup-begin
4  hand right;
5  bin washer in -5,20 : 3 : cm;
6  bin screws in -5,25 : 3 : cm;
7  bin nut;
8  tool screwdriver_phillips in -30,35 : 3 : cm;
9  tool wrench_adjustable in 30,35 : 3 : cm;
10 tool wrench_combination in 33,35 : 3 : cm;
11 tool wrench_combination in 36,35 : 3 : cm;
12 tool hammer_claw in -33,35 : 3 : cm;
13 assembly #1 in -10,-35 : 7 cm;
14 assembly #2 in -30,-32 : 2 : cm;
15 assembly #3 in -10,-10 : 2 : cm;
16 setup-end
17 start
18 take : washer with hand;
19 move : assembly #1 with hand-nondominant
20 from -10,-10 : 3 to 0,0 : 4 : cm;
21 hold : assembly #1 with hand-nondominant;
```

```
22  put : washer with hand in 0,0 : 1 : cm; // Frame #1
23  push : washer with hand in assembly #1; // Frame #2
24  release : assembly #1 with hand-nondominant;
25  take : hammer_claw with hand; // Frame #3
26  take : assembly #1 with hand-nondominant;
27  hold : assembly #1 with hand-nondominant;
28  hit : hammer_claw in 0,0 : 4 : cm; // Frame #4
29  spin : assembly #1 with hand-nondominant;
30  hit : hammer_claw in 0,0 : 4 : cm; // Frame #5
31  turn : assembly #1 with hand-nondominant;
32  release : hammer_claw with hand in 0,0 : 4 : cm;
33  end
34  assemble-end
```

3.5.2. Ejemplo 2

En este ejemplo se desarrolla la descripción mediante el lenguaje de un proceso de ensamble de ruedas de un monopatin que implica herramientas de tipo destornillador y alterna el uso de ambas manos como dominantes.

Listing 3.12: Code example

```
1   assemble-begin
2   example02;
3   setup-begin
4   hand right;
5   assembly #truck in 5::cm;
6   assembly #desk in 0:60:cm;
7   tool hammer_ball_pein;
8   tool clamp;
9   tool pliers_diagonal;
10  tool pliers_long_nose;
11  tool screwdriver_slotted;
12  tool screwdriver_slotted;
13  tool screwdriver_phillips;
14  tool wrench_combination;
15  tool wrench_combination;
16  tool wrench_adjustable;
17  bin bolt in -10:-10:cm;
18  bin nut in -5:-10:cm;
19  setup-end
20  start
21  // Frame01
22  // Initial frame
23  // Frame02
24  take:assembly #truck with hand-nondominant;
25  take:assembly #desk with hand;
26  move:assembly #desk with hand to 10:0:cm;
27  // Frame03
28  take:nut with hand;
29  // Frame04
30  join:assembly #truck with assembly #desk;
31  // Frame05
32  take:nut with hand-nondominant;
33  take:screwdriver_slotted with hand;
34  // Frame06
35  put:nut with hand in assembly #desk;
36  // Frame07
37  take:wrench_combination with hand-nondominant;
38  // Frame08
```

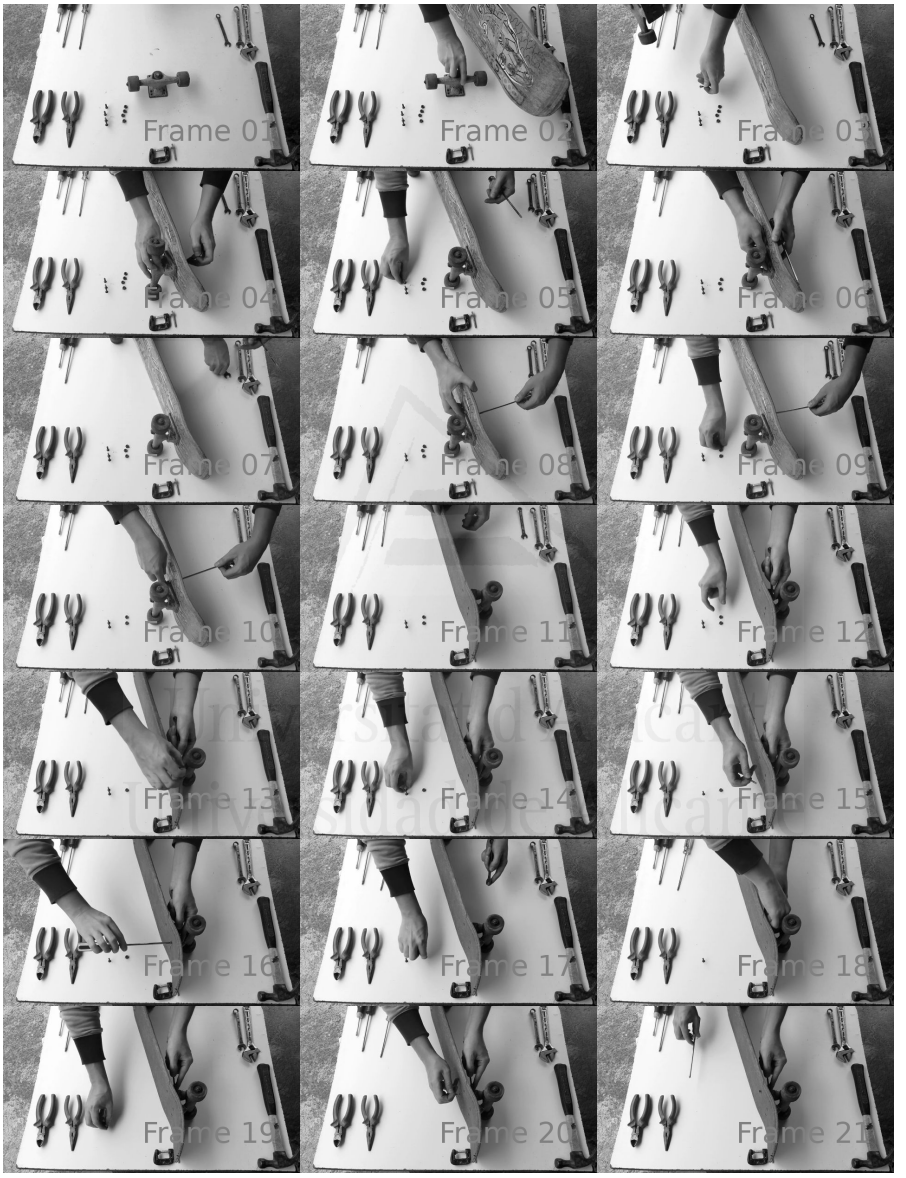


Figura 3.3: Cuadros para ejemplo 2 de MDL

```
39 locknut:wrench_combination with hand-nondominant in assembly #truck;
40 screw:screwdriver_slotted with hand in assembly #desk;
41 // Frame09
42 take:nut with hand-nondominant;
43 take:screwdriver_slotted with hand;
44 // Frame10
45 locknut:wrench_combination with hand-nondominant in assembly #truck;
46 screw:screwdriver_slotted with hand in assembly #desk;
47 // Frame11
48 spin:assembly #desk with any-hand;
49 // Frame12
50 take:nut with hand-nondominant;
51 // Frame13
52 put:nut with hand-nondominant in assembly #desk;
53 // Frame14
54 take:bolt with hand-nondominant;
55 // Frame15
56 put:nut with hand-nondominant in assembly #desk;
57 // Frame16
58 take:screwdriver_slotted with hand-nondominant;
59 // Frame17
60 take:nut with hand-nondominant;
61 // Frame18
62 put:nut with any-hand in assembly #desk;
63 // Frame19
64 take:bolt with hand-nondominant;
65 // Frame20
66 put:bolt with hand-nondominant in assembly #desk;
67 // Frame21
68 take:screwdriver_slotted with hand;
69 end
70 assemble-end
```

3.5.3. Ejemplo 3

En este ejemplo se desarrolla la descripción mediante el lenguaje de un proceso de ensamble de piezas que implica herramientas de tipo sacacorchos y alicate y el uso de ambas manos.

Listing 3.13: Code example

```
35 assemble-begin
36 example03;
37 setup-begin
38 hand right;
39 assembly #truck in 55:-28:cm;
40 assembly #wheel1 in 14:60:cm;
41 assembly #wheel2 in 14:45:cm;
42 assembly #base_plate in 0:5:cm;
43 tool hammer_ball_pein;
44 tool clamp;
45 tool pliers_diagonal;
46 tool pliers_long_nose;
47 tool screwdriver_slotted;
48 tool screwdriver_slotted;
49 tool screwdriver_phillips;
50 tool wrench_combination;
51 tool wrench_combination;
52 tool wrench_adjustable;
53 bin bolt in -50:-30:cm;
```

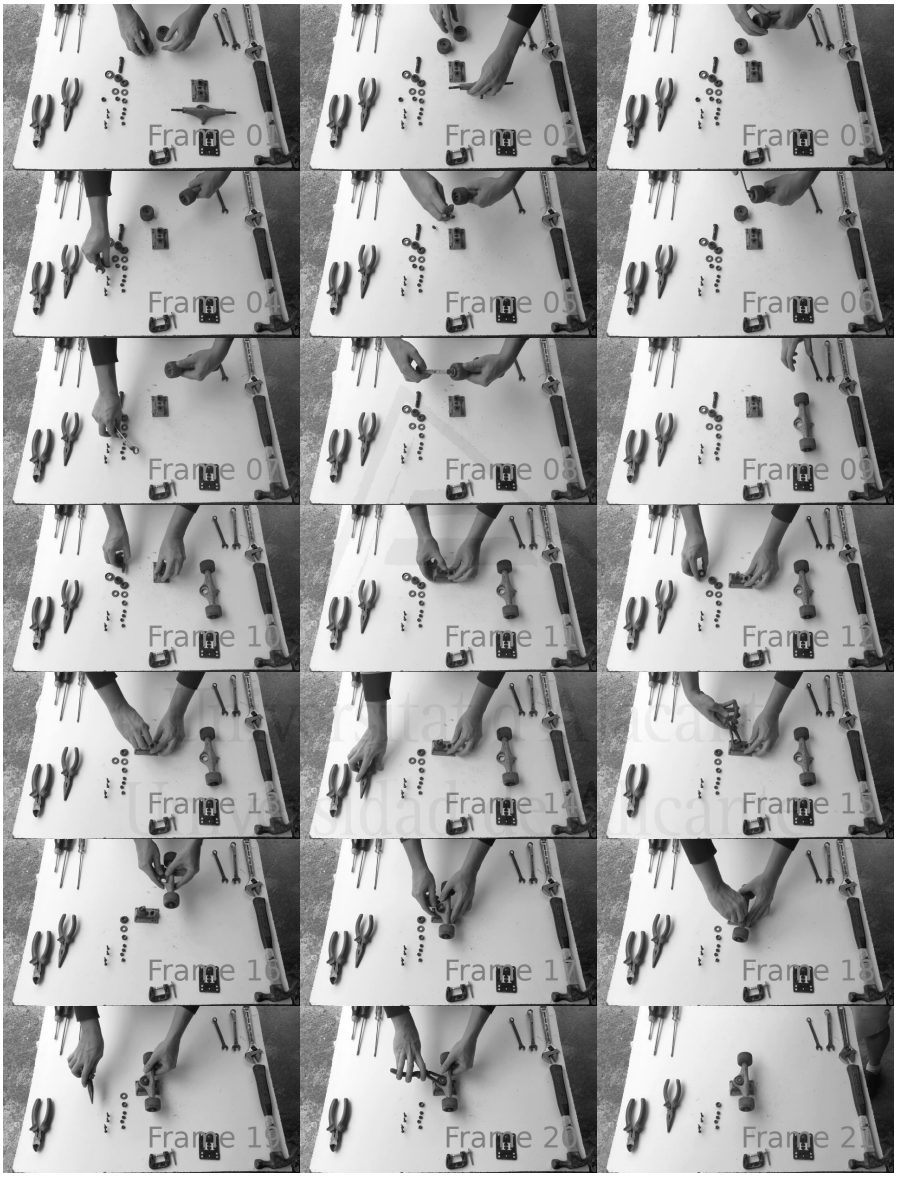


Figura 3.4: Cuadros para ejemplo 3 de MDL

```
54  bin washer in -32:-6:cm
55  bin bolt in -30:30:cm;
56  setup-end
57  start
58  // Frame01
59  take:assembly #wheel1 with hand-nondominant;
60  take:assembly #wheel2 with hand;
61  // Frame02
62  move:assembly #truck with hand to 0:5:cm;
63  // Frame03
64  join:assembly #truck with assembly #wheel1;
65  // Frame04
66  hold:assembly #truck with hand-nondominant;
67  hold:wrench_combination with hand;
68  take:washer with hand;
69  // Frame05
70  put:washer with hand in assembly #truck;
71  // Frame06
72  tight:wrench_combination in assembly #truck;
73  // Frame07
74  take:nut with hand;
75  // Frame08
76  put:nut with hand in assembly #truck;
77  tight:wrench_combination in assembly #truck;
78  // Frame09
79  move:assembly #truck with hand to 0:5:cm;
80  // Frame10
81  take:bolt with hand;
82  take:assembly #base_plate with hand-nondominant;
83  // Frame11
84  join:bolt with assembly #base_plate;
85  // Frame12
86  take:washer with hand;
87  // Frame13
88  put:washer with hand in assembly #base_plate;
89  // Frame14
90  hold:pliers_long_nose with hand;
91  // Frame15
92  hold:pliers_long_nose in assembly #base_plate;
93  // Frame16
94  take:assembly #base_plate with hand-both;
95  // Frame17
96  join:assembly #truck with assembly #base_plate;
97  // Frame18
98  put:nut with hand in assembly #truck;
99  // Frame19
100 hold:pliers_long_nose with hand;
101 // Frame20
102 hold:pliers_long_nose in assembly #base_plate;
103 // Frame21
104 move:assembly #base_plate with hand to 0:0:cm;
105 end assemble-end
```

Figura 3.5: Frames for code example

Listing 3.14: Code example

```
assemble-begin
skateboard;
setup-begin
hand right;
bin washer in -5,20 : 3 : cm;
bin screws in -5,25 : 3 : cm;
bin nut;
tool screwdriver_phillips in -30,35 : 3 : cm;
tool wrench_adjustable in 30,35 : 3 : cm;
tool wrench_combination in 33,35 : 3 : cm;
tool wrench_combination in 36,35 : 3 : cm;
tool hammer_claw in -33,35 : 3 : cm;
assembly #1 in -10,-35 : 7 cm;
assembly #2 in -30,-32 : 2 : cm;
assembly #3 in -10,-10 : 2 : cm;
setup-end
start
take : washer with hand;
move : assembly #1 with hand-nondominant
from -10,-10 : 3 to 0,0 : 4 : cm;
hold : assembly #1 with hand-nondominant;
put : washer with hand in 0,0 : 1 : cm; // Frame #1
push : washer with hand in assembly #1; // Frame #2
release : assembly #1 with hand-nondominant;
take : hammer_claw with hand;           // Frame #3
take : assembly #1 with hand-nondominant;
hold : assembly #1 with hand-nondominant;
hit : hammer_claw in 0,0 : 4 : cm;      // Frame #4
spin : assembly #1 with hand-nondominant;
hit : hammer_claw in 0,0 : 4 : cm;      // Frame #5
turn : assembly #1 with hand-nondominant;
release : hammer_claw with hand in 0,0 : 4 : cm;
end
assemble-end
```

Capítulo 4

Bases de datos de manufactura

En este capítulo se presentan dos bases de datos desarrolladas. La primera base de datos de imágenes de tipo híbrida real-sintético, consta de un conjunto de 29,550 imágenes para entrenamiento de redes neuronales. Estas imágenes fueron generadas usando 591 imágenes reales de herramientas y componentes de uso general en las celdas de manufactura y aplicando diferentes técnicas de aumento de datos.

Además, en este capítulo también se introduce la base de datos de acciones, que contiene las acciones descritas en lenguaje de descripción de procesos de manufactura, propuesto en el Capítulo 3.

Ambas bases de datos fueron desarrolladas como compĺemento al sistema de reconocimiento visual apoyado con un lenguaje de descripción de ensamble.

4.1. Introducción

Para el desarrollo de sistemas inteligentes, la fase de entrenamiento es fundamental para el desarrollo de la solución. En el caso particular de este trabajo, era indispensable que se lograra la identificación de herramientas, piezas en área de trabajo, así como el reconocimiento de la acciones realizadas por los operarios.

Después de realizar la revisión de materiales disponibles para la investigación, se determinó que no se existían fuentes de datos etiquetadas necesarias para realizar el entrenamiento, por lo que se crearon dos bases de datos de uso específicos para procesos manufactura, la primera de imágenes y la segunda de vídeos de acciones.

Las bases de datos propuestas, no solo brindarán soporte para el entrenamientos de redes neuronales para esta investigación, estás también aportan a industria para el entrenamiento de sistemas inteligentes para automatizar tareas.

Como se mencionó en Sección 2.6, la Industria 4.0 tiene una relación muy estrecha con la IA. Su utilización permite mejoras en la productividad, calidad y seguridad en las diferentes fases de producción [7, 47].

Una de las motivaciones para el desarrollo esta base de datos de imágenes es el uso de técnicas de DL para el detección de objetos. Estas arquitecturas requieren de una gran cantidad de datos etiquetados para obtener un rendimiento relevante. Al no encontrar una base de datos pública específica para objetos de manufactura, se procedió a crear una con datos reales y sintéticos mediante etiquetado automático. Se utilizaron herramientas específicas con el fin de simplificar la cantidad de trabajo necesario para el proceso de etiquetado. Estos datos etiquetados son fundamentales para la detección y reconocimiento de objetos en escenas reales [48].

Es importante hacer notar que se realizó un aumentado de datos sintéticos utilizando muestras de objetos reales y entornos de realidad virtual. Estos datos fueron utilizados para entrenar una red YoloV3 [88], cuyo rendimiento fue analizado para comprobar su idoneidad.

En el caso de la base de datos de acciones, su creación fue posible gracias a las grabaciones de un grupo de asistentes de la Universidad de Costa

Rica, que se grabaron realizando operaciones comunes de manufactura, y etiquetaron los datos de forma manual.

El resto del capítulo se estructura de esta manera: se realiza una revisión de las bases de datos de imágenes de uso general más empleadas. Seguidamente se presentan las dos bases de datos generadas: una de imágenes denominada Toolset Dataset y otra de acciones de ensamblajes en manufactura. El capítulo finaliza con la experimentación básica de validación de las bases de datos propuestas.

4.2. Revisión de bases de datos relevantes

En esta sección se revisan tanto las bases de datos públicas de imágenes más relevantes en investigación de detección y reconocimiento de imágenes. Como las bases de datos de vídeos acciones o comportamientos más populares.

4.2.1. Bases de datos de imágenes

En el campo de la detección de objetos, la selección de la base de datos que se usará para el entrenamiento de la red, es un factor clave de éxito ya que determinará el nivel de certeza que tendrá la red una vez entrenada.

Para la realización de este trabajo se realizó una búsqueda exhaustiva para determinar si dentro de las bases de datos disponibles, existe una que se ajuste a los requerimientos para el desarrollo de esta tesis doctoral, entre las bases de datos encontradas que fueran relevantes se encontraron las siguientes:

4.2.1.1. The Pascal Visual Object Classes (VOC)

El Pascal Visual Object Classes (VOC)[1], es una base de datos disponible para su uso público y una competición anual. Además, realizan talleres en forma regular desde el año 2006. La base de datos consiste en 20 categorías cuyas imágenes fueron obtenidas de Flickr[2] [21].

[1] http://host.robots.ox.ac.uk/pascal/VOC/
[2] https://www.flickr.com/

4.2.1.2. ImageNet Large Scale Visual Recognition Challenge

Es principalmente, un desafío de reconocimiento visual llamado ImageNet, es reconocida como una competencia a gran escala [3], ya que contiene 14,197,122 imágenes, 21,841 synsets indexados, organizados según la jerarquía de WordNet.

Este desafío se ha ejecutado anualmente desde 2010 y se ha convertido en el punto de referencia estándar para el reconocimiento de objetos a gran escala. El conjunto de datos publicado contiene un conjunto de imágenes de entrenamiento anotadas manualmente [90].

4.2.1.3. COCO Dataset

El conjunto de datos COCO (Objeto común en contexto) [4] es un conjunto de datos de detección, segmentación y subtitulación de objetos a gran escala. Contiene más de 330 mil fotos de 91 tipos de objetos con un total de 2.5 millones de instancias etiquetadas. Tiene considerablemente más instancias de objeto por imagen en comparación con ImageNet y PASCAL VOC [56].

4.2.1.4. SUN Database

La base de datos SUN (Scene UNderstanding) [5] es un conjunto de datos de categorización de escena. Contiene 131,067 imágenes en 908 categorías de escenas y 313,884 objetos segmentados en 4,479 categorías de objetos. Este conjunto de datos se basa en WordNet [114], tiene imágenes anotadas que cubren una gran variedad de escenas ambientales, lugares y objetos dentro.

4.2.2. Bases de datos de acciones

En esta sección presentamos conjuntos de datos relevantes de reconocimiento de acciones, así como un resumen que destaca la información importante sobre cada uno de ellos. Los conjuntos de datos revisados han

[3] http://www.image-net.org/
[4] http://cocodataset.org/
[5] https://vision.cs.princeton.edu/projects/2010/SUN/

sido seleccionados en función de su innovación en cuanto a la forma en que se adquieren los datos y presentado, su escala, que tiene que ser lo suficientemente grande como para entrenar adecuadamente redes profundas, y los tipos de acciones que incluyen.

4.2.2.1. RGBD-HuDaAct

Esta es una base de datos especializada en reconocimiento de actividad humanas en hogares. Entre las actividades diarias humanas que contiene esta base de datos, se tienen: acostarse, trapear el piso y comer, etc. Utiliza información cámara de video y un sensor de profundidad. [74, 75].

4.2.2.2. UTKinect-Action3D

Esta es una base de datos con información tipo RGB-D y ubicaciones de las articulaciones del esqueleto. Contiene 10 tipos de acción registradas, entre ellas: caminar, sentarse, levantarse, levantar, cargar, lanzar, empujar, tirar, agitar las manos, aplaudir. La base de datos utilizaron 10 sujetos distintos, cada sujeto realiza cada acción dos veces. Entre las características técnicas de los vídeos se tienen: velocidad de fotogramas es de 30 fps, las imágenes RGB tienen una resolución 480x640, la resolución de las imágenes con profundidad es de 320x240. [113].

4.2.2.3. NTU RGB+D

Esta es una base de datos de tipo RGB-D, que contiene 60 clases de acción diferentes, incluidas acciones diarias, comunes y relacionadas con la salud.

La base de datos esta compuesta de de 56,880 RGB-D video, que utilizó 40 sujetos humanos diferentes con un rango de edad de 10 a 35 años. La grabación de los videos se Microsoft Kinect V2. Las secuencias contiene datos de esqueleto (ubicaciones en 3D de 25 articulaciones corporales principales). Los videos fueron tomados con 80 puntos de vista distintos. [57, 94].

4.2.2.4. UCF101

Esta base de datos creada por la Universidad Central de la Florida[6], donde recolectan videos de Youtube[7]. Esta base de datos contiene 101 acciones y 13,320 cortos de video. Se clasifican en cinco categorías: interacción humano-objeto, solo movimiento corporal, interacción humano-humano, tocar instrumentos musicales, deportes. Entre las características técnicas se tienen que se grabaron en entornos no restringidos, incluyen movimiento de cámara, diversas condiciones de iluminación, oclusión parcial, cuadros de baja calidad, etc. Todos los videos tienen una velocidad de cuadro fija y una resolución de 25fps y 320x240. El formato de los videos son de tipo AVI comprimidos usando el códec DivX. [97, 98].

4.2.2.5. StairActions

Esta base de datos contiene 100 categorías de acciones hogareñas cotidianas específicas de diversas tareas hogareñas como enfermería, cuidado y seguridad. Los vídeos fueron obtenidos de diversas fuentes, hay alrededor de 1,000 videos que fueron obtenidos de YouTube o producidos por trabajadores de múltiples fuentes. El número total de videos es 102,462, cuya duración de cada video es entre cinco a seis segundos. [122, 123].

4.2.2.6. EPICKitchens

Esta base de datos participaron 32 participantes, pertenecientes a 10 nacionalidades y todos los videos fueron filmados en sus cocinas. La base de datos consiste de capturas de actividades diarias en la cocina y son secuencias completas de distintas duraciones. Se presentan interacciones típicas con los utensilios de cocina y los electrodomésticos, así como otras tareas como lavar algunos platos en medio de la cocina. Contiene 55 horas de grabación anotadas con tiempos de inicio / finalización para cada acción / interacción, así como los bounding boxes. Los datos fueron capturados usando un Go-Pro colocada en la cabeza del participante. Los videos fueron

[6] www.ucf.edu
[7] www.youtube.com

grabados con un campo de visión lineal (fov), 59.94 fp y resolución de 1920x1080. [14, 15].

4.3. Bases de datos propuestas

Se presenta en esta sección la propuestas de base de datos de imágenes reales y sintéticas, etiquetada para su empleo en los sistemas de detección y reconocimiento de objetos. Así mismo, se describe la base de datos de acciones de ensamblaje grabada y etiquetada para su empleo en la arquitectura de descripción visual de acciones, base de este trabajo.

4.3.1. Base de datos de imágenes "Toolset Dataset"

Para poder llevar a cabo este proyecto de investigación, al igual que con el Manufacturing Description Language, se tuvo que crear una base de datos específica para poder entrenar ANN, se creó una base de datos híbrida compuesta de 24 categorías y las imágenes compuestas entre fotografías reales y aumentadas.

Entre los elementos que contiene la base de datos se tienen: herramientas de uso general para los procesos de manufactura, accesorios de las herramientas y piezas de uso general para ensamble, como lo son tornillos, tuercas, arandelas, y otros más.

Las especificaciones de esta base de datos, fueron pensadas para ser entrenado con YOLO (ver sección 2.9.3), por lo que el etiquetado fue realizado para este fin, tanto para las fotografías reales, como para la imágenes aumentadas.

El ToolSet Dataset fue creada específicamente para satisfacer con especificaciones de la tesis doctoral y usando como base los detalles del lenguaje propuestos llamado Lenguaje de Descripción de Manufactura - Manufacturing Description Language (MDL) (ver sección 3.3), por lo que la categorías de la herramientas corresponden tanto para el lenguaje como para la base de datos.

Para recolección de las imágenes básicas (reales), se buscaron un cada categoría utilizando las bases de datos descritos en la sección Related General Purpose Datasets (ver sección), pero al no encontrar la cantidad

suficiente de las imágenes para las categorías, se hizo una búsqueda en la Web utilizando Google Images®[8] con los se obtuvo una base de 1000 imágenes reales libres de uso, con una resolución intermedia a alta, que fueron usadas posteriormente para aplicarles técnicas de aumento de datos.

En lo referente al proceso de aumento de datos, se aplicaron técnicas de transformación, traslación, cambio de imagen de fondo. Las diferentes transformaciones que generamos las realizamos un total de 50 veces para cada uno de los objetos, utilizando un conjunto de datos conformado por 1000 imágenes para los fondos.

Para generar una base de datos reales, se quiere mucho trabajo por parte una persona, ya que tiene que buscar y etiquetar cada una de la imágenes. Para aumentar el tamaño de la base de datos, se utilizó técnicas computacionales para generar imágenes sintéticas a partir de modelos 3D, con el fin de obtener una mayor variedad de imágenes para la base de datos que satisfacen los requerimientos necesarios para el entrenamiento de la red. Para generar el proceso de creación de imágenes sintéticas se utilizó UnRealEngine que es un conocido motor de videojuegos, que junto con un pluing llamado UnrealROX facilita la tarea de creación de la base de datos sintética, que permitió crear imágenes desde distintos puntos de vista, además se logró realizar el proceso de etiquetado para el formato de YOLO.

Las redes profundas necesitan grandes cantidades de datos etiquetados para obtener niveles aceptables de generalización. La anotación de las imágenes es una tarea que requiere una inversión considerable en la cantidad de tiempo requerido y esfuerzo humano en el proceso.

En la sección anterior se evaluaron las bases de datos de imágenes más relevantes para testear algoritmos de reconocimiento y localización de objetos. Sin embargo, estas bases de datos son generales e incluyen solo unas pocas clases y ejemplos relacionadas con herramientas y accesorios de fabricación.

Con el fin de obtener suficientes datos relevantes y variables para alimentar las arquitecturas profundas, se propuso la generación de una base de datos compuesta de datos reales y sintéticos.

[8]https://images.google.com/

Los datos reales tienen la finalidad de aportar al aprendizaje ruido, reflejos, perturbaciones, y otras características en la adquisición de datos mediante cámaras. Se pretende que la red no llegué a un sobre-aprendizaje debido a la perfección en las capturas obtenidas a través de entornos sintéticos.

Además, dada la dificultad en el proceso de obtención y procesamiento de las imágenes, se han aumentado los datos mediante filtros aplicados a las imágenes como rotaciones y deformaciones.

Los objetos fueron seleccionados considerando su uso en los diferentes procesos de producción. Se compone de un total de 24 herramientas y materiales como: abrazaderas, martillos y tornillos, ejemplos de estos objetos se observan en las figuras 4.1 y 4.3.

4.3.1.1. Imágenes reales

La base de datos está compuesta por imágenes reales con un total de 591 imágenes obtenidas de Internet. Para obtener una cantidad de datos suficiente, que permita a la red ser entrenada adecuadamente, se han aplicado técnicas de un aumento de datos.

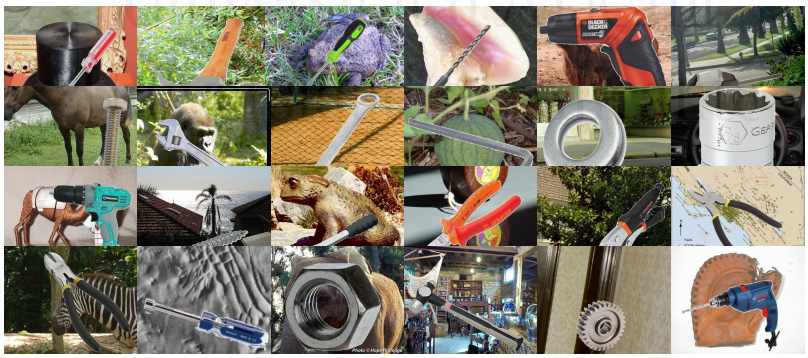


Figura 4.1: Objetos que componen los datos reales con diferentes fondos aleatorios

Este proceso ha sido realizado aplicando diferentes etapas. En primer lugar se han segmentando los objetos que interesan de las imágenes originales. Posteriormente se han aplicado diferentes transformaciones, que incluyen variar el fondo de las imágenes, o aplicar diferentes transformacio-

nes a los objetos de manera aleatoria, tales como: rotaciones, traslaciones, deformaciones y ruido.

Las diferentes transformaciones que generamos, las aplicamos un total de 50 veces para cada uno de los objetos, utilizando un conjunto de datos conformado por 1,000 imágenes para los fondos.

Este proceso permite generar un total de 29,550 nuevas muestras correctamente etiquetas a partir del tamaño de los objetos segmentados (Figura 4.1). Se puede estimar la ubicación del "bounding box" después de aplicar las distintas transformaciones.

4.3.1.2. Imágenes Sintéticas

Los datos reales requieren mucho esfuerzo para la recolección y etiquetado. Además, están limitados a la perspectiva sobre la cual han sido capturadas las imágenes.

Se han generado datos sintéticos a partir de mallas 3D para los diferentes objetos, con el fin de obtener una mayor variedad de puntos de vista de los objetos y aumentar la cantidad de datos disponible.

Figura 4.2: Doce cámaras se extienden alrededor de la zona de generación de los objetos. En este área, es posible variar el fondo y la orientación de los objetos.

Este proceso se realizó utilizando un motor de videojuegos, que cuenta con múltiples plugins. En concreto se empleó UnrealROX [65], un generador que facilita la producción de de datos sintéticos fotorealistas.

$$[label] = [bbox_{xcenter}/W][bbox_{ycenter}/H][bbox_w/W][bbox_h/H] \quad (4.1)$$

Este plugin permite generar diferentes tipos de datos a partir de las simulaciones ejecutadas en el motor, como imágenes RGB, profundidad y máscaras de segmentación. Los tipos de datos con mayor interés para nuestra aplicación son: variaciones en el color y máscaras de segmentación, que se utilizan para generar las etiquetas en el formato definido para YoloV3.

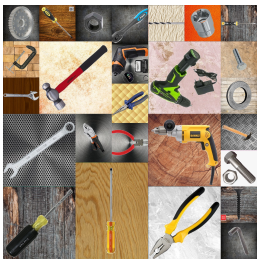
Estas etiquetas se obtienen a partir de la generación de un bounding box con los píxeles máximos y mínimos de la máscara de segmentación. Con estos valores se calcula el centro del bounding box, el ancho y alto de la imagen son normalizados como se muestra en la ecuación 4.1, donde W y H representan el ancho y alto de la imagen.

Figura 4.3: Mallas utilizadas para generar el conjunto de datos sintético.

Para obtener una variabilidad relevante, desplegamos 12 cámaras para representar diferentes puntos de vistas del objeto (Figure 4.2). Además, para evitar que la red memorice el fondo de trabajo y mejore su rendimiento en modo testeo, se realizaron variaciones de manera aleatoria el fondo de las diferentes capturas con 50 muestras diferentes.

El sistema fue preparado para realizar 100 capturas con las cámaras desplegadas para cada objeto. Durante este proceso, se aplicaron de manera aleatoria rotaciones sobre los objetos. Lo que permite obtener varia-

bilidad sobre las muestras y generar un total de 28800 imágenes4.3 con su correspondientes etiquetas del "bounding box" para YoloV3.



(a) Ejemplos de imágenes aumentadas

(b) Ejemplos de las imágenes sintéticas

Figura 4.4: Ejemplos de la base de datos de imágenes "Toolset Dataset"

4.3.2. Base de datos de vídeos de acciones de manufactura

Como se mencionó en la sección 4.3.1, lo que motivó la creación de una base de datos de imágenes de objetos de manufactura, fue la ausencia de una base de datos disponible públicamente para este fin. De igual manera, tampoco se han encontrado bases de datos con vídeos de acciones y comportamientos que contengan acciones específicas de ensambles en manufactura.

La necesidad de disponer de vídeos de ensambles en manufactura muy especializados, que tienen que cumplir con la exigencias de las acciones planteadas en el lenguaje (ver Capítulo 3), se toma la decisión de realizar la grabación y anotación de las siguientes acciones:

Para realizar la grabación de los vídeos, se contó con la ayuda de los asistentes de investigación de la Escuela de Ingeniería Industrial de la Universidad de Costa Rica. Se busco generar variabilidad en las diferentes características personales de los actores: color de la piel, la altura, el uso de ropa de manga larga y manga corta, en diferentes posiciones en la mesa de trabajo y uso de la mano dominante (derecha o izquierda).

Los vídeos utilizados para la validación fueron grabados en un entorno de trabajo con iluminación natural y artificial, con cámaras web de alta definición. La cámara fue ubicada en el centro de la mesa de trabajo de la celda de manufactura, desde un punto de vista cenital.

Tabla 4.1: Acciones para reconocimiento en la base de datos de vídeos

Acción	**Acción en el lenguaje**
Apretar	*tight*
Bloquear tuercas	*locknut*
Atornillar	*screw*
Sostener	*hold*
Golpear	*hit*
Perforar	*drilling*
Liberar	*release*
Poner	*put*

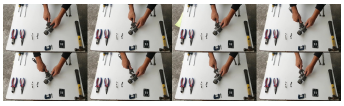
Debido a las limitantes del uso de los laboratorios por la pandemia del `SARS-CoV-2`, las grabaciones fueron realizadas en habitaciones del equipo investigador.

Se utilizaron para las demostraciones de ensambles, distintos monopatines y sus componentes para realizar las grabación de las acciones. Se utilizó una configuración similar a la encontrada en las empresas, donde se tiene una mesa de trabajo con una superficie de un color sólido y claro; y las herramientas colocadas ordenadamente según el producto a ensamblar.

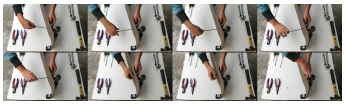
En la Figura 4.2 se puede observar el número de secuencias de los vídeos filmados. También se pueden apreciar las herramientas y las acciones propias que se realizan con cada una de ellas.

Tabla 4.2: Cantidad de secuencias por acción

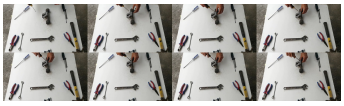
ID	Acción[9]	Secuencias
a	tight	179
b	locknut	30
c	screw	47
d	hold	34
e	hit	46
f	drilling	25
g	release	24
h	put	26



(a) Ejemplo de Llave - apretar



(b) Ejemplo de Llave - bloquear tuercas



(c) Ejemplo de Destornillador - atornillar

(d) Ejemplo de Alicates - sostener

(e) Ejemplo de Martillo - golpear

(f) Ejemplo de Taladro - perforar

(g) Ejemplo de Llave de trinquete - liberar

(h) Ejemplo de Llave de trinquete - poner

Figura 4.5: Ejemplos de acciones con herramientas

4.4. Experimentos

En esta sección, se describen los experimentos realizados para cada una de las bases de datos. En la primera sección se detallará el experimento realizado para validar la base de datos de imágenes. En la última sección, se comenta como fue realizado el experimento para la validación de la base de datos de acciones.

4.4.1. Experimentación para la base de datos sintética de imágenes “Toolset Dataset”

YoloV3 ha sido empleado para validar la detectección de los diferentes objetos que contiene la base de datos, debido a que presenta una alta tasa de aciertos con un tiempo de ejecución muy reducido [88].

Los experimentos realizados con esta red consistieron en utilizar un subconjunto de los objetos disponibles en la base de datos y entrenar la red hasta un máximo de 50,200 épocas. Los objetos utilizados fueron 18 de los 24, los cuales se correspondían con herramientas manuales, como

destornilladores y martillos. No se entreno la red mezclando herramientas manuales junto a materiales como tornillos y arandelas.

mAP	P	R	F_1	IoU
94.6	0.96	0.98	0.97	83.7

Tabla 4.3: La precisión media promedio (mAP), la precisión, recall, F1 score y el promedio en la intersección sobre la unión (IoU) obtenidos como resultado de nuestro entrenamiento

Para llevar a cabo el entrenamiento se mezclaron los datos reales con los sintéticos y se separaron en una proporción 20/80 para generar un conjunto de validación y otro de entrenamiento. A continuación se ajustó el entrenamiento de modo que, además del aumentado de datos "offline" se realizasen transformaciones adicionales en tiempo de entrenamiento, como rotaciones de hasta 40 grados, variaciones en los valores HSV de hasta un 50 % y cambios de escala de hasta un 30 %, partiendo de un tamaño de imagen de 416x416. Además, los hiper-parámetros utilizados para entrenar la red fueron un "learning rate" de 0,001, momentum 0,9 y un "burn in" de 1000.

Los resultados obtenidos al finalizar el entrenamiento pueden observarse en en la tabla 4.3 . En ella podemos observar valores elevados para precisión, recall y F1 score, lo que puede ser un indicio de overffitting en nuestro modelo de entrenamiento. Por eso probamos con objetos adicionales a los utilizados en nuestros conjuntos de validación y test con lo que obtuvimos los resultados de la figura 4.6 . En estas muestras podemos observar como las dos primeras muestras fueron detectadas de manera correcta, aunque en la tercera la red detecta el drill screw como drill gun, debido a que ambas herramientas presentan características similares.

Se unificaron los datos sintéticos junto a los reales y se separaron en una proporción 20/80 para generar un set de validación y otro de entrenamiento. Se ajustó el entrenamiento de modo que, además del aumentado de datos offline. le fueron aplicadas diferentes transformaciones: rotaciones de hasta 40 grados, variaciones en los canales HSV de hasta un 50 % y cambios de escala de hasta un 30 %.

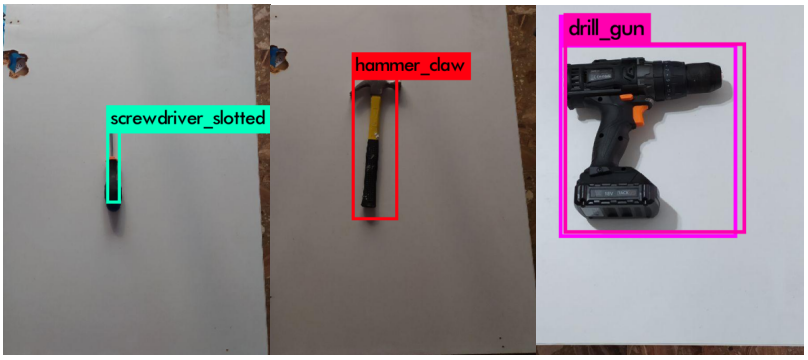


Figura 4.6: Resultados de predicción cualitativa con objetos que no se utilizan en el entrenamiento ni conjuntos de validación

4.4.2. Experimentación para la base de datos de vídeos de acciones de manufactura

Las acciones seleccionadas para integrar la base de datos de acciones se inspiran en las definidas en el lenguaje de descripción de ensambles en manufactura descritas en el Capítulo 3.

El proceso de grabación se realizó utilizando cámaras tipo "web", de alta resolución, estás cámaras fueron posicionadas con una vista superior sobre la mesa de trabajo.

En cuanto a las grabaciones, los sujetos que fueron filmados corresponden a los asistentes de un proyecto de investigación de la Universidad de Costa Rica. Ellos realizaron procesos de ensambles sencillos donde se aplicarán las acciones seleccionadas.

Los vídeos fueron probados utilizando el sistema *ADV* (esta arquitectura se detalla en la sección 2.10) se han aplicado con éxito la detección de acciones observadas en el área de trabajo de la celda de producción.

Capítulo 5

Arquitectura para el control visual de ensambles

En este capítulo se presenta la propuesta de arquitectura computacional para el control visual de ensambles en la Industria 4.0 basado en aprendizaje profundo. La arquitectura permite determinar si el operario está realizando el ensamble de forma correcta y proporcionar información de ayuda a la operación que está realizando. Los sistemas basados en la arquitectura son capaces de percibir el entorno de trabajo (herramientas, partes ensambladas, etc.) así como las acciones que realiza el operario. Las acciones y el entorno representan instrucciones del lenguaje descrito en el Capítulo 3. La utilización del lenguaje y la entrada visual permite la realimentación necesaria al operario.

5.1. Introducción

La Industria 4.0 se ha convertido en un área innovadora que ha generado una evolución acelerada en la forma de cómo las empresas realizan sus labores cotidianas. Una de las principales actividades de una manufactura es el proceso de ensamble. Por su naturaleza, el operario construye distintos productos en su jornada laboral pudiendo confundir las instrucciones de fabricación por diferentes motivos (cansancio, falta de entrenamiento, etc.). En esta tesis doctoral, se especifica una arquitectura computacional basada en visión y aprendizaje profundo. Esta arquitectura permitirá desarrollar sistemas de asistencia de control visual para, por ejemplo, saber si un operador está, por un lado, realizando correctamente el ensamble de un producto y, por otro lado, es capaz de sugerir al operador las acciones necesarias para realizar un determinado ensamble. Todo esto, a través de la monitorización de las actividades de manufactura y del entorno de trabajo.

En esta propuesta se considera la interacción de los humanos con los sistemas robóticos en las industrias automatizadas de una forma más natural, que involucre la seguridad ocupacional, minimizando el estrés a los operarios por el uso de los sistemas robóticos [68]. En conjunto, los colaboradores humanos con las máquinas forman una forma de trabajo sinérgico [35, 36, 50].

La arquitectura (véase la figura 5.1) está compuesta por tres módulos que se detallan en las siguientes secciones: procesamiento visual, procesamiento del lenguaje de descripción de la fabricación y el procesador de comandos de acción. El procesamiento visual es el módulo esencial de la arquitectura que permite percibir el entorno de trabajo y las acciones desarrolladas por el operario para conformar los comandos de acción. Estos comandos especifican de forma conjunta la acción llevada a cabo y la composición del entorno detectado. El módulo de procesamiento del lenguaje de descripción de manufactura permite interpretar las instrucciones de ensamble para especificar un grafo con el que decidir si lo que realiza el operario es correcto.

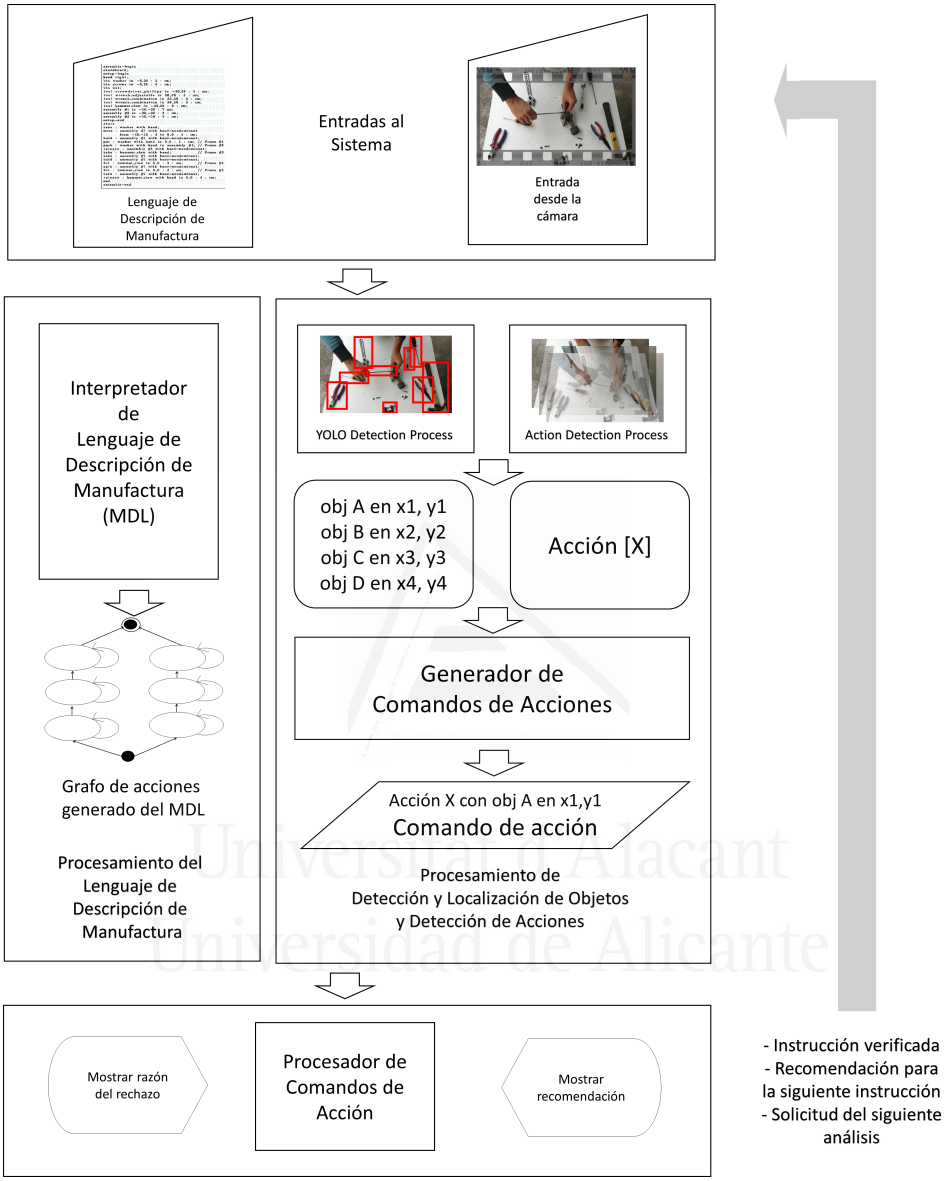


Figura 5.1: Esquema general de la arquitectura propuesta

5.2. Procesamiento visual

El módulo de procesamiento visual es el encargado de realizar una inspección visual del operador y el entorno mientras ensambla las piezas. Para ello, supervisa tanto la herramienta y las piezas utilizadas como las

acciones para llevar a cabo el ensamble. El módulo se compone de dos unidades secuenciales que se detallan a continuación: el reconocimiento del entorno de trabajo y un proceso de detección de acciones. Las salidas de estos módulos se fusionan para proporcionar el *Comando de acción*, que luego se utilizará para evaluar si lo que está haciendo el operador son las instrucciones definidas por el lenguaje.

5.2.1. Reconocimiento del entorno de trabajo

El módulo de reconocimiento del entorno utiliza la secuencia de imágenes que provienen de la captura de vídeo para detectar las herramientas utilizadas en el proceso de ensamble y las piezas (por ejemplo, tuercas, tornillos) que permiten el mismo junto con su ubicación. Aunque la arquitectura propuesta es genérica y podría usarse cualquier detector de objetos, en este documento este módulo se basa en la red neural profunda YOLO (consulte la sección 2.9.3), que le permite realizar una detección en tiempo real con un rendimiento de alta precisión. Un ejemplo de los elementos detectados por el Detector YOLO se pueden ver en la Figura 5.2.

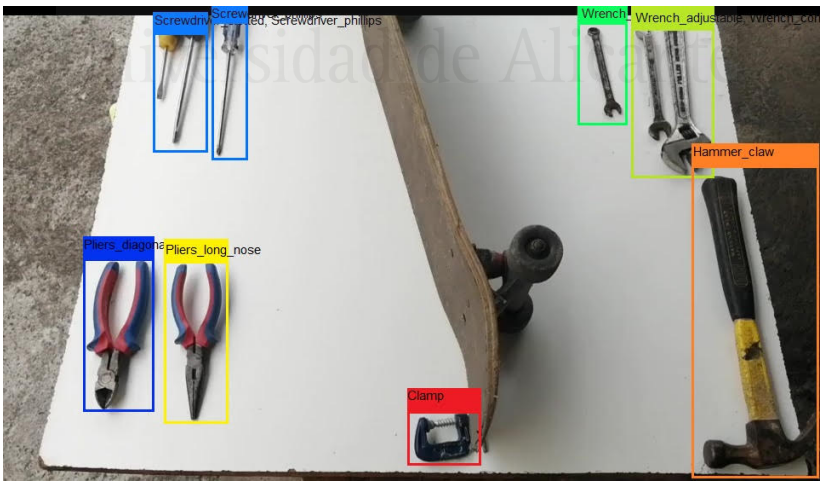


Figura 5.2: Ejemplo de escena de manufactura con la detección de objetos basada en YOLO

5.2.2. Reconocimiento de acciones del operario

Dado que la arquitectura del asistente de control visual podría analizarse principalmente como un sistema de reconocimiento de instrucciones, el sistema debe ser capaz de identificar qué acción está realizando el operador para validar el ensamble adecuado o proponer la siguiente acción a realizar. Este módulo proporciona esta información mediante el análisis de la secuencia de imágenes de vídeo que provienen del módulo de captura.

Este módulo generará la acción como salida, que junto con la salida del módulo de reconocimiento del entorno, explicada anteriormente, formará el *Comando de Acción*, que se utilizará como entrada para la siguiente etapa (Procesador de Comando de Acción). El motor del procesador de acciones se basa en el Deep Activity Description Vector (D-ADV), un descriptor de secuencias de imágenes que puede describir con gran precisión las acciones o actividades que se realizan en la escena [8]. Este descriptor es la variante del *Activity Description Vector* (*ADV*) [6] que se puede usar para métodos de aprendizaje profundo.

El *ADV* y sus variantes se han aplicado con éxito en diferentes áreas para la detección de acciones de individuos y grupos mediante el análisis del movimiento de la persona en su conjunto. En este caso, el descriptor se utiliza para detectar acciones observando el área de trabajo de una celda de producción. El sistema sólo puede ver las herramientas, piezas, componentes y manos del operador. El D-ADV se puede usar con diferentes arquitecturas de red profunda. En este documento, se ha utilizado una arquitectura de dos flujos como se muestra en la Figura 5.3.

Utilizando el vídeo que proviene de la escena del operador en el área de trabajo, la arquitectura procede con el cálculo de la representación de la secuencia de imágenes de la variante profunda del Activity Description Vector (ADV) original, calculando así un cálculo de desplazamiento inicial basado en el flujo óptico. Al mismo tiempo, se estima el primer plano de la escena para calcular la frecuencia a medida que se produce el número de movimientos en una región específica de la escena (este método divide el escenario en regiones de celdas de cuadrícula para discretizar el entorno). Después de calcular el desplazamiento y la frecuencia, se acumulan de acuerdo con un número determinado de imágenes. Además, el

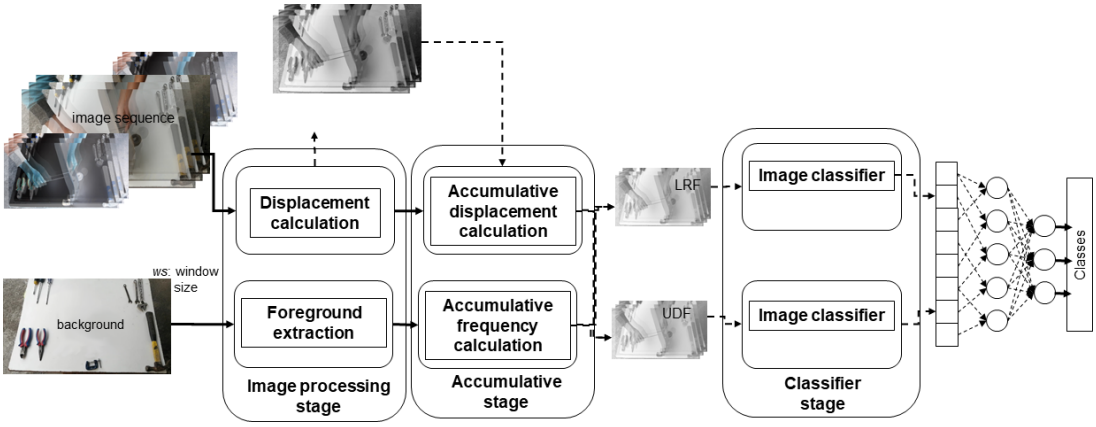


Figura 5.3: Arquitectura de red profunda para la detección de acciones

desplazamiento se separa en los movimientos realizados en cada eje (positivo o negativo) consiguiendo los movimientos hacia arriba (U), hacia abajo (D), hacia la izquierda (L), hacia la derecha (R) y la frecuencia (F)). Finalmente, estos movimientos individuales se concatenan en D-ADV para conformar dos imágenes LRF y UDF que son la entrada de la etapa de clasificación.

La etapa del clasificación es una red de dos flujos compuesta por módulos CNN capaces de clasificar la entrada en múltiples clases. Las salidas individuales se conectan al final de la red mediante una fusión tardía que proporciona la clase final de acción.

5.3. Procesamiento de lenguaje de descripción de fabricación

Este módulo es responsable de generar un grafo de estados con las acciones, incluidas las herramientas que se utilizarán, que debe realizar el operador (ver Fig. 5.4. El lenguaje se creó específicamente para describir las acciones del operador (para más detalles, consulte la sección 3.3). Se puede ver un ejemplo del código en el listado 3.11. En consecuencia, procesa la secuencia de ensamble con lenguaje de instrucciones, generando las salidas esperadas. El grafo de estado como el resultado del procesamiento visual serán la entrada para evaluar las operaciones realizadas en el área

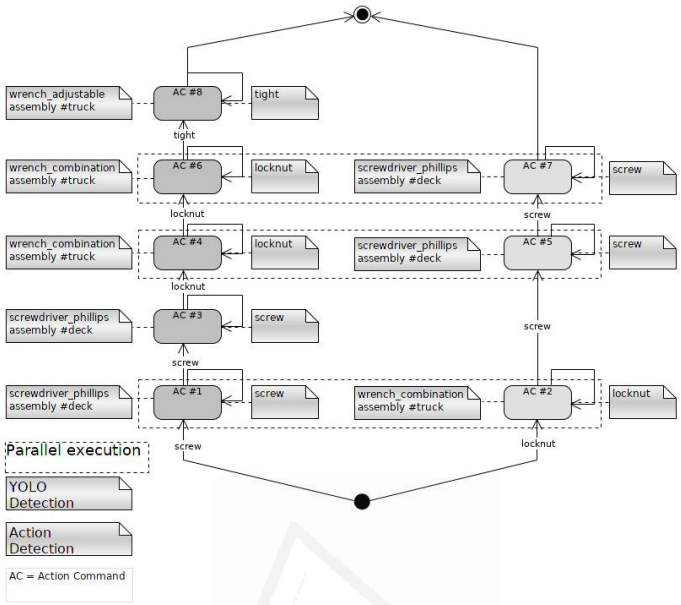


Figura 5.4: Ejemplo del grafo de estados para el procesamiento del lenguaje de descripción de manufactura

de trabajo.

Específicamente, el intérprete de lenguaje de descripción de fabricación (MDLI) se encarga de interpretar las instrucciones descritas por el lenguaje de descripción de fabricación para generar, como resultado del proceso de interpretación, el grafo de estados con los pasos evaluados. Cada nodo representa un estado del ensamble y la arista, el activador para cada cambio de estado a través de las acciones y herramientas para avanzar al proceso de ensamble (ver Fig. 5.5).

Este grafo es el que se utilizará para comparar la salida generada de este módulo y la generada en el procesamiento visual. Permitirá determinar si las instrucciones (acciones y herramientas) se realizaron correctamente, haciendo un seguimiento de la última ejecución realizada. Con este análisis, es posible determinar cuál es la instrucción actual que se ejecutará y la siguiente instrucción que se utilizará como recomendación. En el caso de ambas instrucciones, el intérprete genera las instrucciones en formato de *Comando de Acción*.

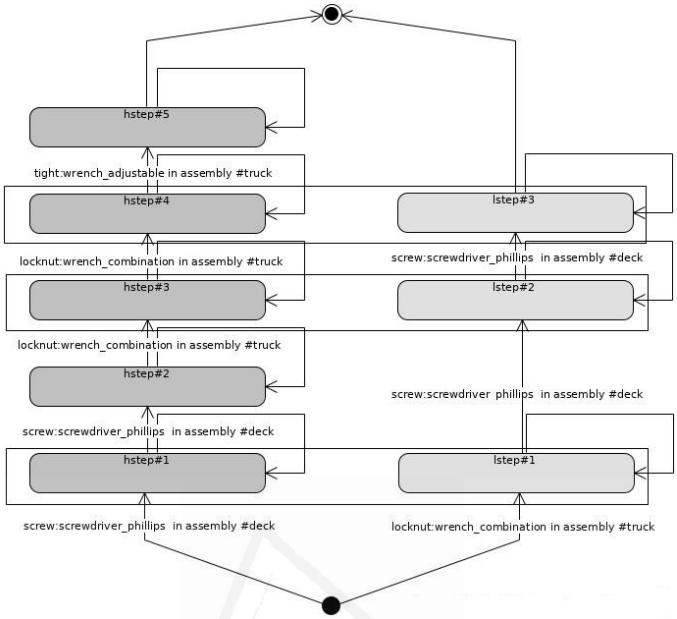


Figura 5.5: Diagrama de estado que representa el lenguaje de fabricación que se procesará para el procesador de comandos de acción

5.4. Procesador de comandos de acciones

Esta es la última etapa de la arquitectura del asistente de control visual y nos permite determinar si el ensamble no se está realizando correctamente o recomendar la siguiente acción a tener en cuenta para ensamblar un producto.

El módulo tiene como entradas el *comando de acción* (objeto detectado en la escena y la acción realizada por el operador) desde el módulo de procesamiento visual y el grafo de estados del procesamiento del lenguaje de descripción de fabricación. El módulo procesa el grafo de estados de acuerdo con el comando de acción para cada imagen del vídeo. En otras palabras, el grafo se recorre a través de las acciones realizadas por los operadores y las herramientas en la escena cuadro a cuadro. Para evitar valores atípicos y errores específicos del módulo de procesamiento visual, el sistema cambia el estado si el cambio a un nuevo estado se basa en el etiquetado de un conjunto de cuadros que proporcionan el mismo resultado o la moda a través de una ventana.

Por lo tanto, si el sistema está diseñado para validar si un ensamble es adecuado, el grafo de estado debe viajar desde el nodo inicial hasta el nodo final si el operador sigue las instrucciones de acuerdo con las especificaciones, si el operario pasa por todos los estados, se considerara que el ensamble fue ejecutado satisfactoriamente. El sistema evaluará en tiempo real las acciones del operario, cada acción equivale un estado del grafo, al realizar la transición de un estado válido (acción reconocida) a otro estado valido, el sistema le indicará al operario que está realizando el ensamble de forma correcta. Si el sistema durante la evaluación, el sistema no reconoce la acción, se considera que es un estado no valido, el sistema esperará una cantidad de segundos (a establecer en el sistema) para esperar que realice la transición a un estado válido, en caso de no lograrse genera una alerta al usuario y bloquea el grafo de ejecución. En caso de que el operador realice una acción que no está planificada en el grafo de estado, el sistema se bloqueará en cualquier nodo, proporcionando una alerta después de un número establecido de segundos (a establecer por el sistema). En caso de que el sistema esté diseñado para recomendar acciones, el sistema proporcionará las diferentes acciones asociadas con las artistas de cada nodo visitado.

5.5. Experimentos

Se han desarrollado un conjunto de experimentos para validar la funcionalidad de la arquitectura propuesta. Concretamente, se define un sistema basado en la misma para comprobar que el operario está realizando de forma correcta el ensamble. Para ello, se han utilizado diferentes ejemplos de procesos de fabricación que, por un lado, se han traducido al lenguaje de descripción de fabricación propuesto y se han convertido en una secuencia compacta de acciones con elementos. Y, por otro lado, se ha grabado unas secuencias en vídeo, anotadas para el conjunto de datos propuesto. Los vídeos han sido procesados por el módulo de procesamiento visual para detectar y reconocer las herramientas y componentes utilizados, y para reconocer la lista de acciones desarrolladas en diferentes marcos. Finalmente, el procesador de comandos de acción se utiliza para comparar tanto la

secuencia de acciones descrita en un grafo de estado como la representación obtenida de nuestra arquitectura de procesamiento de vídeo mediante aprendizaje profundo.

Los experimentos propuestos en esta investigación se realizaron en dos estaciones de trabajo gemelas con Ubuntu 18.04 LTS, Intel (R) Core (TM) i5-8400 2.8GHz 6 núcleos de caché de 9Mb, 32GB de RAM, GeForce RTX (TM) 2070 con 8GB GDDR6 de 256 bits interfaz de memoria, dos discos duros sólidos de 480 GB. Se utilizó TensorFlow 2 (que incluye Keras).

5.5.1. Resultados del procesamiento visual

Para realizar los experimentos del módulo de procesamiento visual, se seleccionó YOLO como arquitectura básica la red para el reconocimiento de objetos, debido a que este tiene un alto rendimiento en la detección de objetos en tiempo real como se puede apreciar en las características descritas del mismo en la sección 2.9.3.

Para el entrenamiento se mezclaron los datos reales con los sintéticos en una proporción 20/80 para generar un conjunto de entrenamiento y otro de validación. Se ajustó el entrenamiento para que el aumento de datos aplicara transformaciones adicionales en tiempo de entrenamiento, tales como: rotaciones de hasta 40 grados, variaciones en los HSV de hasta un 50 % y cambios de escala de hasta un 30 %, partiendo de un tamaño de imagen de 416. Los hiperparámetros utilizados para entrenar la red fueron un learning rate de 0,001, momentum 0,9 y un *burn-in* de 1000.

Un ejemplo de la identificación de los objetos en uno de los vídeos de validación se muestra en la figura 5.2. En ella se distinguen los cuadros delimitadores de las herramientas utilizadas, para cada uno de ellos el correspondiente identificador de clase.

En cuanto a la validación del procesador de reconocimiento de acciones. Se utilizó una validación cruzada para determinar los conjuntos de entrenamiento y prueba obteniendo los resultados que se pueden ver en la tabla 5.1. Se evaluaron las 8 acciones básicas, que se tomaron como el núcleo básico descrito en el lenguaje. Hay acciones que son muy diferentes de las demás, como perforar y golpear que se obtuvo el 100 % de sensibilidad durante la ejecución de las pruebas. Hay otras acciones cuya ejecución con las

herramientas es muy similar, pero tienen ligeras variaciones con respecto a la forma en que el operador realiza la acción. Estas acciones son: apretar, poner, liberar que tiene un valor aproximado al 98 % de sensibilidad.

Para las 8 acciones evaluadas, 5 acciones logran una sensibilidad superior al 98 %, obteniendo un rendimiento muy alto al reconocer las acciones. Para las 3 acciones restantes (retener, contratuerca y atornillar), los valores de sensibilidad fueron 94.65 %, 95.28 % y 96.47 % proporcionando nuevamente valores de muy alto rendimiento.

Las figuras 5.6 y 5.7 muestran un análisis de estos datos utilizando la curva ROC. Estas métricas de evaluación permiten verificar el rendimiento del modelo de clasificación gráficamente. Como puede verse en los valores tanto del análisis de las secuencias completas como de cada imagen de la secuencia, prácticamente hay valores muy cercanos a 1 (valor máximo) como verdaderos positivos, lo que indica que las pruebas realizadas tienen altos niveles de confianza en términos de reconocimiento de la acción.

Tabla 5.1: Resultados del reconocimiento de acciones

Class	**Frame**		**Sequence**	
	Sensitivity	**Specificity**	**Sensitivity**	**Specificity**
Hold	94,65 %	94,83 %	97,06 %	99,19 %
Tight	98,79 %	98,76 %	99,44 %	99,12 %
Screw	96,47 %	96,45 %	100,00 %	100,00 %
Locknut	95,28 %	95,18 %	98,33 %	96,82 %
Hit	100,00 %	100,00 %	100,00 %	100,00 %
Drilling	100,00 %	99,99 %	100,00 %	100,00 %
Put	97,89 %	97,88 %	100,00 %	100,00 %
Release	98,06 %	98,14 %	100,00 %	100,00 %
Overall	**96,37 %**	**97,94 %**	**99,36 %**	**99,42 %**

5.5.2. Resultado del procesador de comandos de acciones

La validación del procesador de comandos de acción se realizó aplicando los principios de un analizador sintáctico. Se trata de verifica que las acciones del usuario (provenientes del núcleo de reconocimiento de acciones) en forma de comandos, se consideren como los tokens para evaluar de

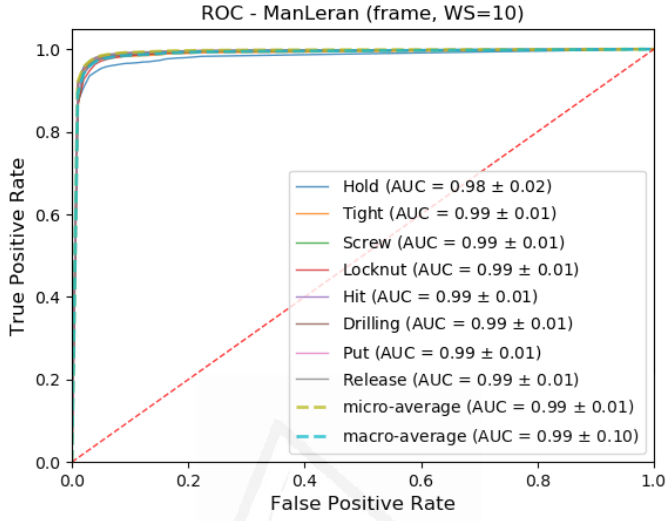


Figura 5.6: Resultados ROC de los cuadros de imágenes

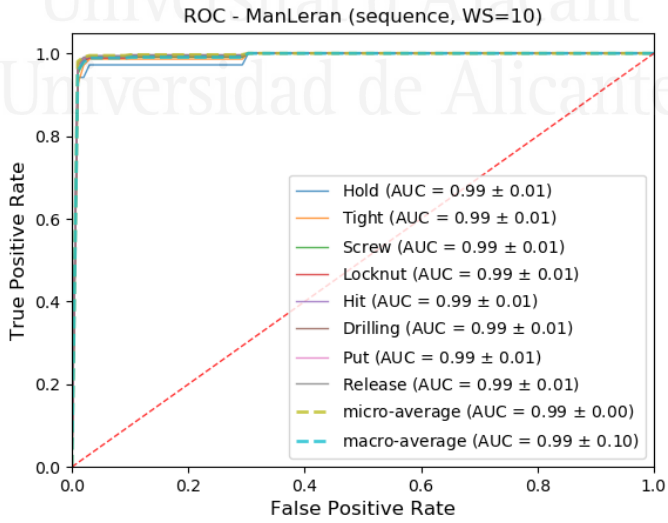


Figura 5.7: Resultados ROC de las secuencias de imágenes

la gramática del lenguaje. Es importante detallar que lo que llamaremos gramática en este punto son las instrucciones descritas por los ingenieros de calidad o producción y escritas usando el Lenguaje de Descripción de Fabricación.

En esta primera parte del comando de acción, se tomaron los vídeos de prueba y se generó una tabla con las acciones identificadas. Esta es la entrada junto con las instrucciones *tokenizadas* que permiten generar un diagrama de estado, que se encarga de verificar que la transición entre los estados (un frame que contiene una acción reconocida se considera como un estado).

El vídeo se revisa en tiempo real para determinar en qué punto cambia el estado. Este estado se compara con el diagrama de estado de las instrucciones. Si se realiza la transición, el proceso se considera .ejecución correcta". El proceso finaliza cuando todas las transiciones de estado se completan de acuerdo con las definiciones hasta llegar al estado final, la condición de ejecución se cambia a .ejecución completa".

Todos los vídeos probados, generaron con éxito los diagramas de estado. Y por lo tanto, fue posible generar la condición de .ejecución completa". El diagrama de estado nos permitió establecer qué cambios de estado eran válidos para hacer las recomendaciones.

La figura 5.8 muestra un ejemplo de este módulo. Se puede considerar en los resultados que el módulo identifica las acciones en el momento correcto del vídeo, incluso es interesante notar que el Procesador de reconocimiento de acciones en la acción de atornillar, detecta un poco antes de lo que el vídeo está etiquetado como atornillar . Esto se debe a que el vídeo se considera como atornillar hasta el momento en que gira el destornillador, en cambio el sistema de detección de acciones lo determina desde el momento en que la punta del destornillador toca el tornillo. Además, se muestra cómo cambian los estados a medida que se identifican en el tiempo, estas acciones son generadas en tiempo real por el Procesador de reconocimiento de acciones. Este ejemplo se generó a partir de uno de los vídeos de validación, donde se utiliza la línea de tiempo generada por el Procesador de reconocimiento de acciones y el gráfico, se marca en gris oscuro a medida que las acciones se ejecutan con el tiempo hasta que se

completa la ejecución.

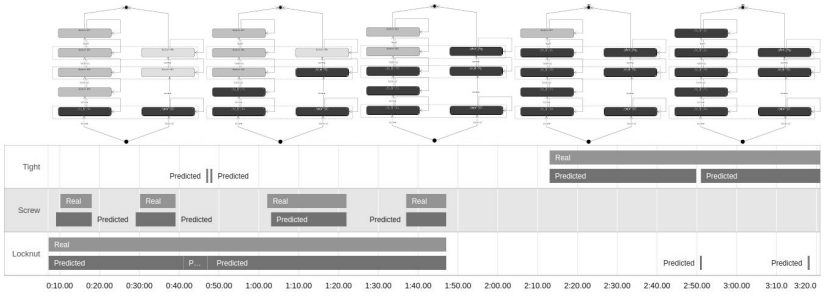


Figura 5.8: Ejecución de gráfico de acciones con líneas de tiempo de reconocimiento de acciones.

Capítulo 6

Conclusiones

En este capítulo final, se detallan las conclusiones del trabajo, se enumeran las aportaciones principales de la tesis, presentando las publicaciones en revistas y congresos internacionales derivadas y finalmente se proponen algunas líneas de investigación futura.

6.1. Conclusiones

En este capítulo final de la investigación de la tesis doctoral se pregunta una sinopsis de cada uno de los capítulos que componen en este documento, para luego finalizar con las conclusiones generadas en la investigación.

En el capítulo 2, llamado “Estado del arte”, se hace un resumen de los conceptos relacionados que le da sustento a la investigación. Además, que identificar aquellas necesidades que fueron identificadas en la formulación de la problemática. Inicialmente las primeras definiciones de la problemáticas a resolver giraban en entorno a las mejoras de la producción en las Pequeñas y Medianas Empresas en Costa Rica. Por lo que se investigo sobre las Ingeniería Industrial, y las áreas en que se ve involucrada esta ciencia. Se identificaron las necesidades que tienen este tipo de empresas, que mayoritariamente realizan trabajo manual, necesitan para realizar la migración a entornos con sistemas inteligentes.

Durante esta fase de recaudación de información, se identifica que cerca el 78 % de las PyME tiene un colaborador con conocimiento básicos de producción, generando que apliquen técnicas de Lean manufacturing. Se identifica que estas empresas utilizan técnicas de registro de información de como se fabrican sus productos a través de flujogramas. En este capítulo también se hace una revisión del estado del arte, de las técnicas de la Inteligencia artificial (IA) aplicada en la manufactura. Esto genera tres taxonomías, una de la aplicación de las tecnologías de IA, mecanismos de IA para la interacción con los operarios y la IA en las técnicas de navegación dentro de la fabricas.

Se hace una revisión de los sistemas de reconocimiento y localización de objetos, donde se evalúan las arquitecturas que según la revisión bibliográficas son las que presentan mayores niveles de exactitud en sus resultados y con altos niveles de rendimiento. Se hace un resumen sobre las técnicas de identificación de acciones, de las principales características que deben tener en cuanto al reconocimiento y predicción de acciones. Como esta extracción utilizan técnicas de representación tipo esqueleto. Para concluir la revisión de este capítulo se hace un análisis del Image Captioning, que explora las técnicas de como a través de la visión por computadora un

sistema inteligente puede describir las acciones que se están ejecutando en la escena.

Para el capítulo 3, se introduce la primera propuesta original de este trabajo, el Lenguaje de Descripción de procesos de Manufactura (MDL), para realizar esta propuesta se evaluaron la técnica Therbligs, que es un estudio de micromovimientos y del AML/X de IBM que fue un lenguaje de los ochenta, que IBM utilizaba para describir procesos para robots. En lo referente al MDL se presenta una síntesis de los elementos de las gramáticas, que permiten realizar la definición de ejemplos. Se presentan ejemplos de la vida real creados con MDL.

En el capítulo 4, se muestra el resultados de dos bases de datos desarrolladas en esta investigación como insumo para poder entrenar la arquitectura de control visual para ensambles.

La primera base de datos es de imágenes. Que consta de imágenes reales, aumentadas y sintéticas de elementos comunes en la manufactura, incluye un conjunto de herramientas básicas, todas ellas etiquetadas. En este punto se considera que sería conveniente incrementar la diversidad de herramientas electro-mecánicas, como tornos, cepilladores de rotación entre otros. También explorar la opción de generar estaba base de datos en un datos RGB-D.

La segunda base de datos, consiste de vídeos de acciones comunes, estas secuencias de vídeos se grabaron según las especificaciones del MDL. Esta base de datos fue gradaba usando vídeos en alta definición en estaciones de trabajo desde un punto de vista superior. Todos los vídeos han sido etiquetados. Además, igual se podría usar información de tipo RGB-D.

En el capítulo 5, se presenta la arquitectura para el control visual de ensambles. En esta solución contiene tres módulos principales:

1. *Módulo de procesamiento del lenguaje de descripción de manufactura*, este módulo se encarga de la interpretar las instrucciones de las tareas que debe realizar el operario, estas instrucciones fueron escritas utilizando el lenguaje propuesto en esta tesis doctoral. Este módulo genera como salida un grafo de dos "vías", una para cada mano de las tareas que tiene que ir ejecutando el operario.

2. *Módulo de procesamiento visual*, este módulo es el encargado de procesar las entradas visuales que provienen de la cámara que esta observando el área de trabajo. Genera como salida un "comando de acción", que es una unidad lógica que identifica que acción esta realizando el operario, los objetos presentes en el área de trabajo y la ubicación de cada uno de los objetos a partir del centro del área de trabajo. Las acciones las identifica con un componente especializado en detección de acciones, que se basa en "D-ADV". El componente que identifica el objeto y la ubicación del mismo, utiliza la arquitectura YOLO.

3. *Módulo de procesamiento de comandos de acción*, esté módulo es el núcleo de sistemas de control visual, ya que toma las entradas de los módulos anteriores para generar el análisis, se puede ver en dos componentes, que se detallan a continuación:

   a) Componente de análisis de comandos de acción, acá se toma el grafo generado por el Módulo de procesamiento del lenguaje de descripción de manufactura, para usarlo como el "mapa" que debe cursar el operario para construir el producto. Lleva el control, de las estados por los cuales se ha ejecutado el grafo, para conocer el estado actual de la ejecución, esto lo logra con el comando de acción que proviene del módulo de procesamiento visual.

   b) Componente de sistema de recomendaciones, una vez evaluado el grafo respecto al comando de acción que ha recibido y determinar el estado actual. El sistema puede evaluar cuales acciones futuras se pueden ejecutar. Por ejemplo si hay un cambio de herramienta el sistema puede notificar al operario como sugerencia o si se logra determinar cual componente se requiere podría notificarlo de alguna forma, como una luz led que parte debe tomar el operario.

En este sección de se exponen las conclusiones, donde se analizarán las principales aportaciones realizadas a través de tres propuestas que trabajar en forma sinérgica para generar la arquitectura propuesta. La primera

propuesta donde se detalla un lenguaje creado propiamente para este trabajo de investigación. Se expone dos bases de datos que fueron construidas para esta investigación: la primera base de datos, consiste en un conjunto de imágenes reales y sintéticas de objetos comunes para la manufactura, la segunda base de datos contiene una selección de vídeos con acciones realizadas en entornos de producción. Ambas bases de datos, son necesarias para el entrenamiento de la redes de detección de objetos y acciones. Se presenta la última propuesta, presenta los elemento relevantes de la arquitectura para el control visual de ensamblajes, donde se detalla los componentes que lo constituyen, los cuales son: el componente procesador del lenguaje, el componente de procesamiento visual y el componente final que realiza el procesamiento y análisis de los componentes iniciales para determinar si el operario esta realizando las tareas correctamente y que genera las sugerencias al operario. Finalmente se incluye algunos los trabajos a futuro de corto, mediano y largo plazo que se pueden derivar de esta investigación.

6.2. Aportaciones

En este trabajo se han realizado 4 aportaciones principales en el campo de la industria 4.0, y más concretamente en la aplicación de técnicas de visión e inteligencia artificial en ensamblajes de productos en industrias manufactureras. Más concretamente las aportaciones consisten en:

- **Taxonomías del aprendizaje automático aplicadas la Industrial 4.0**, se llevó a cabo la revisión sistemática del estado del arte. Para identificar dentro del contexto de la Cuarta Revolución Industrial (Industria 4.0) aquellos elementos de la visión por computadora y métodos de interacción humano - computadora que están siendo utilizados en las fabricas. Generando como productos taxonomías de la interacción humano - robot, aplicaciones del aprendizaje automático en la industria, técnicas de intercambio de información y autonomía utilizando aprendizaje profundo.

- **Lenguaje de Descripción de Manufactura - Manufacturing**

Description Language (MDL), se desarrollo un lenguaje de representación visual de tareas, cuyo uso principal objetivo es la descripción de las tareas que lleva a cabo un operador en las industrias para completar un proceso de ensamblaje. Se trata de un lenguaje de uso dual: puede ser utilizado independiente para documentar procesos y como complemento del control visual de ensambles para registro de las acciones a verificar. También es la base de conocimiento para que el sistema inteligente pueda sugerir las recomendaciones de las siguientes tareas al operario.

- **Base de datos de imágenes "Toolset Dataset"**, se generó una base de datos compuesta de 24 categorías de herramientas, accesorios y partes, comúnmente utilizados en en las estaciones de trabajo de entornos de manufactura. Esta base de datos de imágenes etiquetada es para uso específico de manufactura que puede ser utilizada para los sistema de aprendizaje máquina. Las imágenes utilizadas para esta base de datos son fotografías reales de los elementos anteriores, imágenes aumentadas a partir de los fotogramas reales, así como imágenes generadas sintéticamente. Todas las imágenes fueron etiquetadas en formato YOLO.

- **Base de datos de vídeos de acciones de manufactura**, En línea con la aportación anterior, se ha generado una base de datos compuesta de vídeos etiquetados que representan acciones requeridas para el entrenamiento de sistemas inteligentes para la detección de tareas de manufactura.

- **Diseño de una arquitectura inteligente para el control visual de ensambles**, se trata de la aportación fundamental de la tesis en la que se especifica una arquitectura capaz de monitorizar el entorno de trabajo y al operario para asistirle en el proceso de manufactura. Los sistemas basados en la arquitectura implementan redes de aprendizaje profundo para la detección de las herramientas, piezas, componentes y acciones en secuencias de vídeo de la estación de trabajo. La entrada visual es comparada con el lenguaje propuesto para determinar si cumple la secuencia de producción y/o para sugerir al

operario las acciones siguientes a llevara cabo.

6.3. Publicaciones derivadas

Dentro de las aportaciones que generó estas investigación doctoral en el área de la Interacción humano - computadora, Visión por computadora aplicada específicamente a los sistemas de control en la manufactura, Bases de datos para el entrenamiento visual de redes neuronales y Uso de lenguajes formales en la cuarta revolución industrial. Referente a estos puntos se han realizado publicaciones en los siguientes congresos 14th International Work-Conference on Artificial Neural Networks (IWANN 2017), International Journal of Computer Vision and Image Processing, 2017 (IJCVIP 2017), 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020) y Computers in Industry (CI)

A continuación se presenta un resumen de cada uno de los trabajos publicados:

- **Machine Learning Improves Human-Robot Interaction in Productive Environments: A Review** (IWANN 2017)

  [Publicado]

- **Automatic Learning Improves Human-Robot Interaction in Productive Environments: A Review** (IJCVIP 2017)

  [Publicado]

- **Manufacturing Description Language for Process Control in Industry 4.0** (SOCO 2020)

  [Aceptado]

- **ToolSet: A Real-Synthetic Manufacturing Tools and Accessories Dataset** (SOCO 2020)

  [Aceptado]

- **Deep Learning-Based Visual Control Assistant for Assembly in Industry 4.0** (CI)

[Enviado]

6.4. Trabajos futuros

Aunque se ha conseguido el diseño e implementación de una arquitectura funcional para el control visual de ensamblajes que incluye tanto un lenguaje de descripción como bases de datos propias diseñadas para alimentar dicha arquitectura. Diversos problemas y extensiones pueden plantearse como trabajos futuros:

- Mejoras en el lenguaje para conseguir una traducción más directa a la descripción visual de las acciones.
- Extensión de la base de datos de imágenes de herramientas y accesorios para hacerla mucho más amplia y que incluya muchos más entornos reales y variabilidad.
- Extensión de la base de datos de videos de acciones para que incluya muchas más acciones en muchos más entornos, con ensambles realizados por diferentes personas y en diferentes sectores de aplicación.
- Mejora de las arquitecturas profundas empleadas tanto en la detección y reconocimiento de objetos que podría incluir pose 6D, como en la arquitectura de reconocimiento de acciones que podría reconocer acciones de bajo nivel.

Lista de Acrónimos

ADV *Activity Description Vector*

ANN *Artificial Neural Networks*

LCE *Assembly Control Language*

BD *Big Data*

CAD *Computer-aided design*

FPS *Cuadros por segundo*

Industria 4.0 *Cuarta Revolución Industrial*

SC *Cyber-Physical Systems*

D-ADV *Deep Activity Description Vector*

DL *Deep Learning*

DNN *Diffusion-Neural Network*

FR-CNN *Fast R-CNN*

FrR-CNN *Faster R-CNN*

FPV *First Person Visiona*

CFM *Flexible Manufacturing Cell*

IHR *Human-Robot Interaction*

II *Ingeniería Industrial*

IA *Inteligencia Artificial*

IoT *Internet of Things*

LSTM *Long Short-Term Memory*

ML *Machine Learning*

MiPyME *Micro, Pequeñas y Medianas Empresas*

PyME *Pequeñas y Medianas Empresas*

CNN *Redes neuronales convolucionales*

RNN *Redes neuronales recurrentes*

RvNN *Redes neuronales recursivas*

RetinaNet *RetinaNet (Focal loss)*

RGB *Rojo, Verde y Azul*

RGB-D *Rojo, Verde y Azul con Profundidad*

SLAM *Simultaneous Localization and Mapping*

SSD *Single Shot MultiBox Detector*

SPP-net *Spatial Pyramid Pooling*

CV *Visión por Computadora*

YOLO *You Only Look Once*

YOLO 9000 *You Only Look Once - 9000*

YOLOv2 *You Only Look Once - versión 2*

YOLOv3 *You Only Look Once - versión 3*

Bibliografía

[1] Abdelhameed, W. (2019). Industrial Revolution Effect on the Architectural Design. *2019 International Conference on Fourth Industrial Revolution, ICFIR 2019*, pages 1–6. 28, 32

[2] Abellan-Abenza, J., Garcia-Garcia, A., Oprea, S., Ivorra-Piqueres, D., and Garcia-Rodriguez, J. (2019). Classifying Behaviours in Videos with Recurrent Neural Networks. *Deep Learning and Neural Networks*, pages 965–980. 46

[3] Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer USA. 37, 44

[4] Ahmad, R. and Plapper, P. (2016). Safe and Automated Assembly Process using Vision Assisted Robot Manipulator. *Procedia CIRP*, 41:771–776. 25

[5] Arce Brenes, J. A. and León Segura, G. (2019). Estado situacional de la pyme en costa rica, serie 2012-2017. Technical report, Ministerio de Economía, Industria y Comercio. Dirección General de Apoyo a la Pequeña y Mediana Empresa http://reventazon.meic.go.cr/informacion/estudios/2019/pyme/INF-012-19.pdf. 4

[6] Azorin-Lopez, J., Saval-Calvo, M., Fuster-Guillo, A., and Garcia-Rodriguez, J. (2016). A novel prediction method for early recognition of global human behaviour in image sequences. *Neural Processing Letters*, 43(2):363–387. 98

[7] Bahrin, M. A. K., Othman, M. F., Azli, N. N., and Talib, M. F. (2016). Industry 4.0: A review on industrial automation and robotic. *Jurnal Teknologi*, 78(6-13):137–143. 77

[8] Borja-Borja, L. F. (2020). Arquitectura de visión y aprendizaje para el reconocimiento de actividades de grupos usando descriptores de movimiento. *Doctoral Thesis, University of Alicante*. 98

[9] CÁMARA DE COMERCIO DE COSTA RICA (2015). ESTADÍSTICAS ECONÓMICAS. Technical report, CÁMARA DE COMERCIO DE COSTA RICA, San José, Costa Rica. 4

[10] Canal, G., Escalera, S., and Angulo, C. (2016). A real-time Human-Robot Interaction system based on gestures for assistive scenarios. *Computer Vision and Image Understanding*, 149:65–77. 25

[11] Cao, C.-Y., Zheng, J.-C., Huang, Y.-Q., Liu, J., and Yang, C.-F. (2019). Investigation of a Promoted You Only Look Once Algorithm and Its Application in Traffic Flow Monitoring. *Applied Sciences*, 9(17):3619. 45

[12] Cheng, Y., Tao, F., Zhao, D., and Zhang, L. (2015). Modeling of manufacturing service supply-demand matching hypernetwork in service-oriented manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 45:59–72. 26

[13] Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., and Fraisse, P. (2016). Collaborative manufacturing with physical human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13. 24

[14] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The EPIC-KITCHENS dataset. *CoRR*, abs/1804.02748. 82

[15] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2020). The epic-kitchens dataset: Collection, challenges and baselines. 82

[16] Ding, W., Gu, J., Shang, Z., Tang, S., Wu, Q., Duodu, E. A., and Yang, Z. (2017). Semantic recognition of workpiece using computer vision for shape feature extraction and classification based on learning databases. *Optik - International Journal for Light and Electron Optics*, 130:1426–1437. 24

[17] Drath, R. and Horch, A. (2014). Industrie 4.0: Hit or hype? [Industry Forum]. *IEEE Industrial Electronics Magazine*, 8(2):56–58. 32

[18] Erdin, M. E. and Atmaca, A. (2015). Implementation of an Overall Design of a Flexible Manufacturing System. *Procedia Technology*, 19:185–192. 23, 24, 32

[19] Ericson, Gary; Franks, Larry; Rohrer, B. (2016). How to choose algorithms for Microsoft Azure Machine Learning. 31, 33

[20] ESRI (2020). How single-shot detector (SSD) works? 43

[21] Everingham, M., Eslami, S. M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2014). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136. 78

[22] Farsal, W., Anter, S., and Ramdani, M. (2018). Deep learning: An overview. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, SITA'18, New York, NY, USA. Association for Computing Machinery. 40

[23] Fast-Berglund, Å., Fässberg, T., Hellman, F., Davidsson, A., and Stahre, J. (2013). Relations between complexity, quality and cognitive automation in mixed-model assembly. *Journal of Manufacturing Systems*, 32(3):449–455. 7, 30, 33

[24] Ferguson, D. (2000). Therbligs: The Keys to Simplifying Work. 52

[25] Ferreiro, S. and Sierra, B. (2012a). Comparison of machine learning algorithms for optimization and improvement of process quality in conventional metallic materials. *International Journal of Advanced Manufacturing Technology*, 60(1-4):237–249. 7, 30, 32, 35

[26] Ferreiro, S. and Sierra, B. (2012b). Comparison of machine learning algorithms for optimization and improvement of process quality in conventional metallic materials. *International Journal of Advanced Manufacturing Technology*, 60(1-4):237–249. 38

[27] for Quality, A. S. (2020a). *Quality Glossary - Q*. 3

[28] for Quality, A. S. (2020b). *What is Lean?* 8

[29] Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C. B., Wang, C. C., Shin, Y. C., Zhang, S., and Zavattieri, P. D. (2015). The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69:65–89. 26

[30] Garrell, A., Villamizar, M., Moreno-Noguer, F., and Sanfeliu, A. (2017). Teaching Robot's Proactive Behavior Using Human Assistance. *International Journal of Social Robotics*. 25

[31] Gong, M. and Shu, Y. (2020). Real-Time Detection and Motion Recognition of Human Moving Objects Based on Deep Learning and Multi-Scale Feature Fusion in Video. *IEEE Access*, 8:25811–25822. 43

[32] Goodrich, M. A. and Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3):203–275. 34

[33] Groover, M. P. (2007). *Work Systems and the Methods, Measurement, and Management of Work*. Pearson Education, Inc. 52

[34] Guo, L., Hao, J.-h., and Liu, M. (2014). An incremental extreme learning machine for online sequential learning problems. *Neurocomputing*, 128:50–58. 35

[35] Hedelind, M. and Jackson, M. (2011). How to improve the use of industrial robots in lean manufacturing systems. *Journal of Manufacturing Technology Management*, 22(7):891–905. 7, 9, 13, 29, 30, 31, 33, 52, 95

[36] Hermann, M., Pentek, T., and Otto, B. (2016). Design principles for industrie 4.0 scenarios. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016-March:3928–3937. 9, 13, 29, 31, 95

[37] Herrero, H., Moughlbay, A. A., Outón, J. L., Sallé, D., and de Ipiña, K. L. (2017). Skill based robot programming: Assembly, vision and Workspace Monitoring skill interaction. *Neurocomputing*, 0:1–10. 24

[38] Hornung, A., Bennewitz, M., and Strasdat, H. (2010). Efficient vision-based navigation. *Autonomous Robots*, 29(2):137–149. 7, 30, 33, 35

[39] Institute of Industrial and Systems Engineers (2019). The Industrial Engineering Body of Knowledge (IEBoK). , (January):1–42. 6, 21, 22

[40] Jiang, S., Chen, G., Song, X., and Liu, L. (2019). Deep patch representations with shared codebook for scene classification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 15(1s):1–17. 40

[41] Kanawaty, G. (2008). *Introducción al estudio del Trabajo*. EDITORIAL LIMUSA DE C.V., 11 edition. 8, 22

[42] Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., Kim, B. H., and Noh, S. D. (2016). Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 3(1):111–128. 32

[43] Khandelwal, R. (2019). SSD : Single Shot Detector for object detection using MultiBox. 43

[44] Khazri, A. (2019). Faster RCNN Object detection. 42

[45] Kong, Y. and Fu, Y. (2018). Human action recognition and prediction: A survey. 46

[46] Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Niebles, J. C. (2017). Dense-Captioning Events in Videos. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:706–715. 47

[47] Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering*, 6(4):239–242. 77

[48] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. 77

[49] Lee, J., Bagheri, B., and Jin, C. (2016). Introduction to cyber manufacturing. *Manufacturing Letters*, 8:11–15. 7, 30, 33

[50] Lee, J., Bagheri, B., and Kao, H.-A. (2014). Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics. *Int. Conference on Industrial Informatics (INDIN) 2014*, (November 2015). 9, 13, 29, 31, 35, 95

[51] Lee, J. H. and Ha, S. H. (2009). Recognizing yield patterns through hybrid applications of machine learning techniques. *Information Sciences*, 179(6):844–850. 35

[52] Leng, J. and Jiang, P. (2015). A deep learning approach for relationship extraction from interaction context in social manufacturing paradigm. *Knowledge-Based Systems*, 100:188–199. 26, 35

[53] Leo, M., Medioni, G., Trivedi, M., Kanade, T., and Farinella, G. M. (2015). Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15. 34, 35, 38

[54] Li, D. C. and Yeh, C. W. (2008). A non-parametric learning algorithm for small manufacturing data sets. *Expert Systems with Applications*, 34(1):391–398. 35

[55] Li, J., Gu, J., Huang, Z., and Wen, J. (2019). Application research of improved YOLO V3 algorithm in PCB electronic component detection. *Applied Sciences (Switzerland)*, 9(18). 45

[56] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755. 79

[57] Liu, J., Shahroudy, A., Perez, M. L., Wang, G., Duan, L.-Y., and Kot Chichung, A. (2019). Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1. 80

[58] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:21–37. 43

[59] Luo, R. C., Hsu, Y. T., Wen, Y. C., and Ye, H. J. (2019). Visual image caption generation for service robotics and industrial applications. *Proceedings - 2019 IEEE International Conference on Industrial Cyber Physical Systems, ICPS 2019*, pages 827–832. 47

[60] Lv, X., Dai, C., Chen, L., Lang, Y., Tang, R., Huang, Q., and He, J. (2020). A robust real-time detecting and tracking framework for multiple kinds of unmarked object. *Sensors (Switzerland)*, 20(1). 44

[61] López, J. A. (2007). *Modelado de sistemas para visión de objetos especulares. Inspección visual automática en producción industrial.* Doctoral thesis, University of Alicante, Spain. 4

[62] M. Tim Jones (2017). Deep learning architectures – IBM Developer. 39

[63] Makris, S., Karagiannis, P., Koukas, S., and Matthaiakis, A. S. (2016). Augmented reality system for operator support in human???robot collaborative assembly. *CIRP Annals - Manufacturing Technology*, 65(1):61–64. 7, 30, 33

[64] Mancini, M., Karaoguz, H., Ricci, E., Jensfelt, P., and Caputo, B. (2018). Kitting in the Wild through Online Domain Adaptation. *IEEE International Conference on Intelligent Robots and Systems*, pages 1103–1109. 48, 54

[65] Martinez-Gonzalez, P., Oprea, S., Garcia-Garcia, A., Jover-Alvarez, A., Orts-Escolano, S., and Garcia-Rodriguez, J. (2020). Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 24(2):271–288. 86

[66] Martinez-Martin, E. and del Pobil, A. P. (2017). Robust Motion Detection and Tracking for Human-Robot Interaction. *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, pages 401–402. 25

[67] Mehlmann, G., Häring, M., Janowski, K., Baur, T., Gebhard, P., and André, E. (2014). Exploring a Model of Gaze for Grounding in Multimodal HRI. *Proceedings of the 16th International Conference on Multimodal Interaction - ICMI '14*, pages 247–254. 35

[68] Meisner, E., Isler, V., and Trinkle, J. (2008). Controller design for human-robot interaction. *Autonomous Robots*, 24(2):123–134. 9, 13, 29, 35, 95

[69] Melzner, J. and Bargstadt, H. J. (2016). A framework for 3D-model based job hazard analysis. *Proceedings - Winter Simulation Conference*, 2016-Febru:3184–3185. 28

[70] Mohammad, Y. and Nishida, T. (2009). Toward combining autonomy and interactivity for social robots. *AI and Society*, 24(1):35–49. 35

[71] Monostori, L. (2002). AI and machine learning techniques for managing complexity, changes and uncertainties in manufacturing. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 15(1):119–130. 24, 35

[72] Nackman, L. R., Lavin, M. A., Highsmith Taylor, R., Dietrich, W. C., and Grossman, D. D. (1986). AML/X: a programming language for design and manufacturing. *ACM '86: Proceedings of 1986 ACM Fall joint computer conference*. 53

[73] Nguyen, A., Do, T.-T., Reid, I., Caldwell, D. G., and Tsagarakis, N. G. (2019). V2cnet: A deep learning framework to translate videos to commands for robotic manipulation. 47, 54

[74] Ni, B., Gang Wang, and Moulin, P. (2011a). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1147–1153. 80

[75] Ni, B., Gang Wang, and Moulin, P. (2011b). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1147–1153. 80

[76] Okubo, N., Nara, K., Takemura, S., and Ueda, Y. (2016). Applying an instructional design process to development of an independent verification and validation training program. *Proceedings - 2016 IEEE 29th Conference on Software Engineering Education and Training, CSEEandT 2016*, pages 237–240. 28

[77] Ong, K., Haw, S.-C., and Ng, K.-W. (2019). Deep Learning Based-Recommendation System. In *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems*, pages 6–11, New York, NY, USA. ACM. 38

[78] Panait, Luke, Panait, L., and Luke, S. (2005). Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 3(11):387–434. 35

[79] Park, S. and Kim, D. (2019). Study on 3D action recognition based on deep neural network. *ICEIC 2019 - International Conference on Electronics, Information, and Communication*, pages 5–7. 46

[80] Patten, T., Vincze, M., Kropatsch, W., Wien, T. U., Fraundorfer, F., Roth, P. M., Schenk, F., Thalhammer, S., and Park, K. (2019). SyDD: Synthetic Depth Data Randomization for Object Detection using Domain-Relevant Background. *24thComputer Vision Winter Workshop*, pages 14–22. 4, 13

[81] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (2018). A Survey on Deep Learning. *ACM Computing Surveys*, 51(5):1–36. 38, 39, 40

[82] Priore, P., De La Fuente, D., Puente, J., and Parreño, J. (2006). A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. *Engineering Applications of Artificial Intelligence*, 19(3):247–255. 27, 35

[83] Puik, E., Telgen, D., van Moergestel, L., and Ceglarek, D. (2017). Assessment of reconfiguration schemes for Reconfigurable Manufacturing Systems based on resources and lead time. *Robotics and Computer-Integrated Manufacturing*, 43:30–38. 23, 24, 32

[84] Quesada, G., Quesada, O., Jara, E., and Arias, A. (2013). Estado de situaciÓn de las pymes en costa rica. 2013. Technical report, Gabriela QuesadaÓscar QuesadaErick JaraAndrea Arias y Ministerio de Economía, Industria y Comercio. http://reventazon.meic.go.cr/informacion/estudios/2013/pyme/indicadores/informe.pdf. 4

[85] Ramík, D. M., Madani, K., and Sabourin, C. (2014). A Soft-Computing basis for robots??? cognitive autonomous learning. *Soft Computing*, 19(9):2407–2421. 35

[86] Rani, P., Liu, C., Sarkar, N., and Vanman, E. (2006). An empirical study of machine learning techniques for affect recognition in human-robot interaction. *Pattern Analysis and Applications*, 9(1):58–69. 35

[87] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788. 44

[88] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. 77, 89

[89] Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149. 42

[90] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252. 79

[91] Sader, S., Husti, I., and Daróczi, M. (2019). Quality management practices in the era of industry 4.0. *Zeszyty Naukowe Politechniki Częstochowskiej Zarządzanie*, 35. 3

[92] Santoro, M., Marino, D., and Tamburrini, G. (2008). Learning robots interacting with humans: From epistemic risk to responsibility. *AI and Society*, 22(3):301–314. 8, 30, 31, 33, 35

[93] Schr??ter, D., Kuhlang, P., Finsterbusch, T., Kuhrke, B., and Verl, A. (2016). Introducing Process Building Blocks for Designing Human Robot Interaction Work Systems and Calculating Accurate Cycle Times. *Procedia CIRP*, 44:216–221. 24

[94] Shahroudy, A., Liu, J., Ng, T., and Wang, G. (2016). Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019. 80

[95] Shi, J., Chang, Y., Xu, C., Khan, F., Chen, G., and Li, C. (2020). Real-time leak detection using an infrared camera and Faster R-CNN technique. *Computers & Chemical Engineering*, 135:106780. 41

[96] Siddique, N. H., Mitchell, R., O'Grady, M., and Jahankhani, H. (2011). Cybernetic approaches to robotics. *Paladyn*, 2(3):109–110. 10, 29

[97] Soomro, K., Zamir, A. R., and Shah, M. (2012a). UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402. 81

[98] Soomro, K., Zamir, A. R., and Shah, M. (2012b). Ucf101: A dataset of 101 human actions classes from videos in the wild. 81

[99] Starke, J., Chatzilygeroudis, K., Billard, A., and Asfour, T. (2019). On Force Synergies in Human Grasping Behavior. *IEEE-RAS International Conference on Humanoid Robots*, 2019-October:72–78. 57

[100] Sudha, L., Dillibabu, R., Srivatsa Srinivas, S., and Annamalai, A. (2016). Optimization of process parameters in feed manufacturing using artificial neural network. *Computers and Electronics in Agriculture*, 120:1–6. 35

[101] Sun, D. W. (2016). *Computer Vision Technology for Food Quality Evaluation: Second Edition*. Elsevier. 37, 38

[102] Tang, H., Peng, A., Zhang, D., Liu, T., and Ouyang, J. (2020). SSD real-time illegal parking detection based on contextual information transmission. *Computers, Materials and Continua*, 62(1):293–307. 44

[103] Tatic, D. and Tesic, B. (2017). The application of augmented reality technologies for the improvement of occupational safety in an industrial environment. *Computers in Industry*, 85:1–10. 34, 35

[104] Tsai, T. I. and Li, D. C. (2008). Utilize bootstrap in small data set learning for pilot run modeling of manufacturing systems. *Expert Systems with Applications*, 35(3):1293–1300. 26, 35

[105] Universidad Politécnica deValencia (2018). Therbligs. 52

[106] Vlassis, N., Toussaint, M., Kontes, G., and Piperidis, S. (2009). Learning Model-free robot control by a Monte Carlo em algorithm. *Autonomous Robots*, 27(2):123–130. 35

[107] Wang, L., Qiao, Y., and Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:4305–4314. 34

[108] Wang, L., Schmidt, B., and Nee, A. Y. C. (2013). Vision-guided active collision avoidance for human-robot collaborations. *Manufacturing Letters*, 1(1):5–8. 7, 30, 33, 35

[109] Wang, X., Chen, W., Wu, J., Wang, Y. F., and Wang, W. Y. (2018). Video Captioning via Hierarchical Reinforcement Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4213–4222. 47

[110] Warren, M. (2016). Using Job Instruction: It's more than just for training. 27

[111] Witold, P. and Shyi-Ming, C. (2020). *Deep Learning Architectures*, volume 866. Springer. 41

[112] Wu, Z., Jiang, Y. G., Wang, X., Ye, H., and Xue, X. (2016). Multi-stream multi-class fusion of deep networks for video classification. *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*, pages 791–800. 40, 41

[113] Xia, L., Chen, C., and Aggarwal, J. K. (2012). View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. 80

[114] Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. (2016). SUN Database: Exploring a Large Collection of Scene Categories. *International Journal of Computer Vision*, 119(1):3–22. 79

[115] Xiao, S., Wang, Z., and Folkesson, J. (2015). Unsupervised robot learning to predict person motion. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 691–696. 8, 30, 33, 35

[116] Xu, J., Wang, W., Wang, H., and Guo, J. (2020). Multi-model ensemble with rich spatial information for object detection. *Pattern Recognition*, 99:107098. 41, 42, 44

[117] Yan, J., Yan, S., Zhao, L., Wang, Z., and Liang, Y. (2019). Research on Human-Machine Task Collaboration Based on Action Recognition. *Proceedings - 2019 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering, SMILE 2019*, pages 117–121. 46

[118] Yang, Y., Cai, Z., Yu, Y., Wu, T., and Lin, L. (2019). Human action recognition based on skeleton and convolutional neural network. In *2019 Photonics Electromagnetics Research Symposium - Fall (PIERS - Fall)*, pages 1109–1112. 46

[119] Yang, Y., Li, Y., Fermüller, C., and Aloimonos, Y. (2015). Robot learning manipulation action plans by "watchingünconstrained videos from the World Wide Web. *Proceedings of the National Conference on Artificial Intelligence*, 5:3686–3692. 46, 48, 54

[120] Yao, T., Pan, Y., Li, Y., Qiu, Z., and Mei, T. (2017). Boosting Image Captioning with Attributes. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:4904–4912. 47

[121] Yeen Gavin Lai, N., Hoong Wong, K., Halim, D., Lu, J., and Siang Kang, H. (2019). Industry 4.0 enhanced lean manufacturing. In *2019 8th International Conference on Industrial Technology and Management (ICITM)*, pages 206–211. 8

[122] Yoshikawa, Y., Lin, J., and Takeuchi, A. (2018a). STAIR actions: A video dataset of everyday home actions. *CoRR*, abs/1804.04326. 81

[123] Yoshikawa, Y., Lin, J., and Takeuchi, A. (2018b). Stair actions: A video dataset of everyday home actions. 81

[124] Zhou, L., Cao, S., Liu, J., Tan, T., Du, F., Fang, Y., and Zhang, L. (2018). Design, manufacturing and recycling in product lifecycle: New challenges and trends. *4th IEEE International Conference on Universal Village 2018, UV 2018*, 2019-Janua:1–6. 28, 29, 32