

Proyecto de Grietas en Madera

Documento de Requisitos del Sistema (DRS)

Autor: Andrés Tena De Tena

Autor: Javier Albaráñez Martínez

Modelo: IEEE/ANSI 830-1998

Índice

1	Introducción	2
1.1	Propósito del documento	2
1.2	Alcance del producto	2
1.3	Definiciones, acrónimos y abreviaturas	2
1.4	Referencias	2
1.5	Descripción del resto del documento	2
2	Descripción general.....	3
2.1	Perspectiva del producto	3
2.2	Funciones del producto	3
2.3	Características del usuario	3
2.4	Restricciones generales	3
2.5	Suposiciones y dependencias	3
3	Requerimientos específicos.....	4
4	Apéndice.....	5
5	Índices	6
5.1	Palabras clave.....	6
5.2	Bibliografía.....	6
5.3	Webs	6
5.4	Empresas	6

1 Introducción

En esta sección se proporcionará una introducción a todo el documento. Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento. Además, se proporciona un enlace al repositorio de GitHub donde se ha realizado el proyecto: https://github.com/AlbaranezJavier/AIVA_2021_AJ.

1.1 Propósito del documento

Documento dirigido a José Vélez Serrano, perteneciente a una empresa dedicada a la industria de la madera. El objetivo de este documento es acotar el riesgo analizando los requisitos del proyecto.

1.2 Alcance del sistema

Se desea crear un método de programación que sea integrable con el sistema informático de la empresa. De ahora en adelante, el método a desarrollar será nombrado como “crack_detector”.

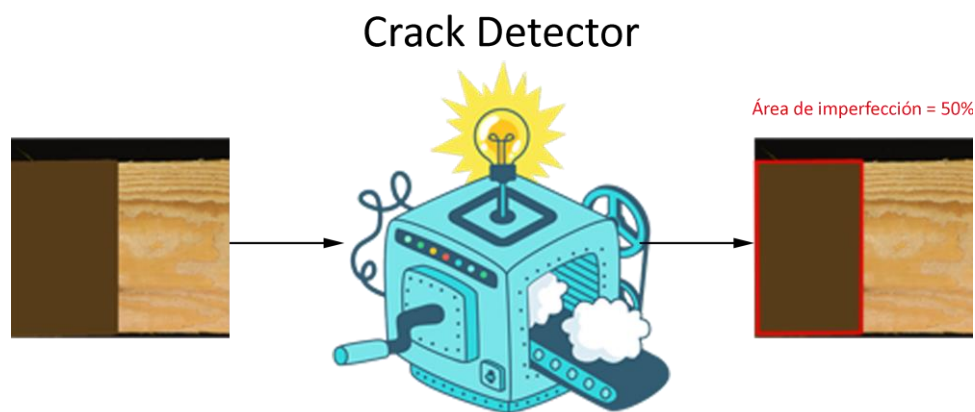


Figura 1. - Esquema simple [referencia].

El objetivo de crack_detector es detectar el tamaño de las imperfecciones (como nudos, fendas, etc) en un trozo de madera de un tipo determinado y cuantificar el porcentaje que este supone del total. Lo que no va a hacer este método es clasificar si la pieza analizada es defectuosa o no, tampoco dispondrá de una interfaz de usuario.

Disponer de crack_detector le va a permitir a la empresa analizar cada pieza de madera de una forma más eficiente, reduciendo el número de lotes defectuosos y repercutiendo en un ahorro económico.

1.3 Definiciones, acrónimos y abreviaturas

Fenda. - Se denomina fenda a toda separación de las fibras (raja o hendidura) en dirección longitudinal.

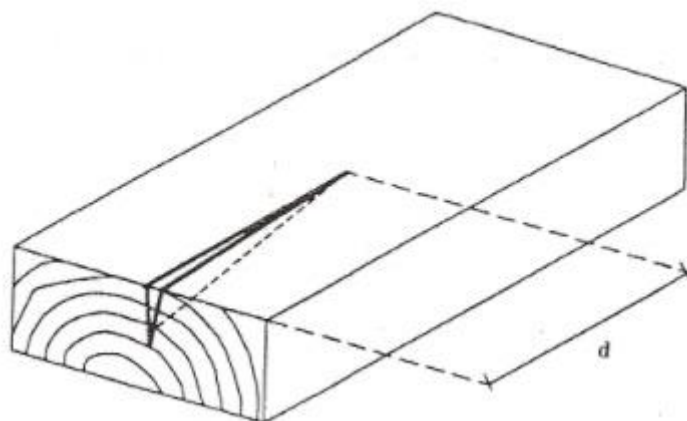


Figura 2. - Ejemplo de fenda.

Fenda pasante. - Se trata de una fenda que se extiende entre dos superficies opuestas.

Nudo. - Se le define como el área de tejido leñoso resultante del rastro dejado por el desarrollo de una rama.

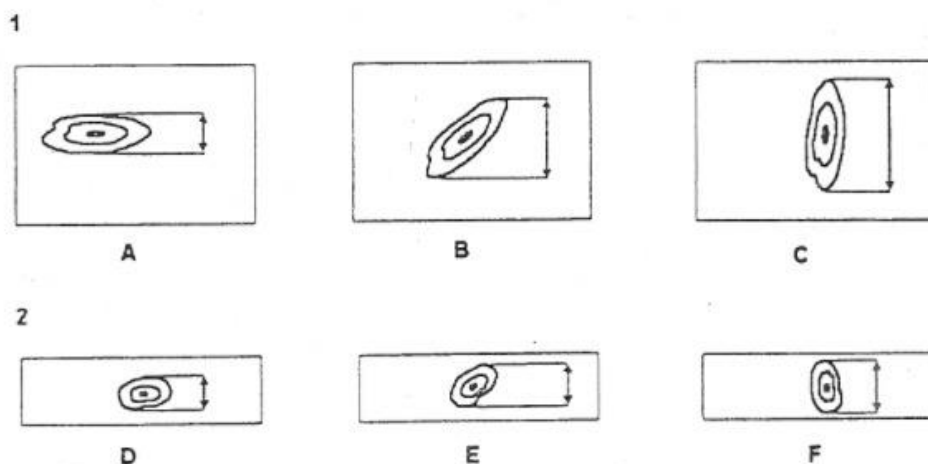


Figura 3. - Ejemplo de nudos, 1 nudos de cara y 2 nudos de canto.

Norma Europea UNE EN 1310.- Método de medida de los diferentes nudos del estándar UNE (Una Norma Española).

1.4 Referencias

- Información sobre la madera [[permalink](#)]
- Aplicación VIA para etiquetar muestra [[web](#)]

1.5 Descripción del resto del documento

En la siguiente sección se describen todos aquellos factores que afectan al producto y a sus requisitos. Finalmente, en la sección 3 se describen los requisitos con más detalle.

2 Descripción general

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

2.1 Perspectiva del producto

La idea es desarrollar una aplicación que permita determinar el tamaño de las imperfecciones en piezas de madera.

Para esto se desarrollará un método en Python que cargará las imágenes utilizando OpenCV y que podrá ser llamado desde Java.

2.2 Funciones del producto

Este método recibirá una imagen a color (ver figura 3 y figura 4), la cual contendrá distintas imperfecciones, estas podrán estar repartidas por toda la pieza, por tanto, el método realizará una búsqueda de estas devolviendo la longitud de cada una de ellas.



Figura 4. - Ejemplo sin imperfecciones.



Figura 5. - Ejemplo con imperfecciones.

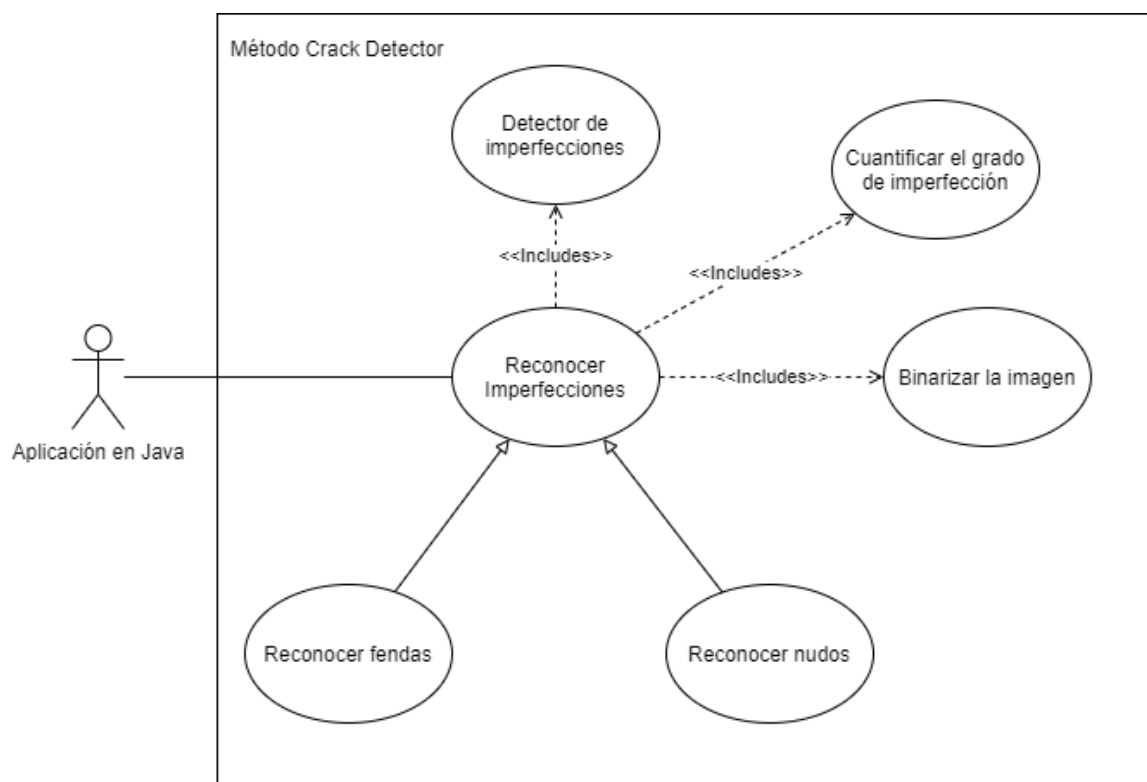


Figura 6. - Casos de uso.

2.3 Características del usuario

Desarrollador que use el método desde un programa en Java para construir un sistema más general.

2.4 Restricciones generales

Descripción de alto nivel de los requisitos no funcionales.

- Lenguaje de programación. El lenguaje que se utilizará será Python, ya que este dispone de bibliotecas que nos facilitarán el trabajo, además es uno de los lenguajes más extendidos para la realización de proyectos en visión artificial.
- Limitaciones del hardware. Preguntar en la próxima reunión al cliente.
- Protocolos de comunicación. Llamadas a métodos desde Java.
- Criticidad de la aplicación. En caso de no reconocer imperfecciones en la pieza de madera, esta se clasificaría como apta y por tanto aumentará el número de piezas defectuosas en los lotes.
- Requisitos a los usuarios. Estos deberán tener conocimientos de programación ya que deberán integrar el método en su sistema.

2.5 Suposiciones y dependencias

- Describir las suposiciones sobre las que se basan los requisitos y que si cambian pueden afectarles.
- Para el desarrollo del método se dispondrá de 100 imágenes cedidas por la empresa, cada una de estas contendrá distintos tipos de imperfecciones.
- La detección de las imperfecciones se realizará sobre la imagen en escala de grises.
- Al digitalizar la imperfección esta deberá tener una resolución de al menos un pixel de ancho en toda su longitud.
- Se desarrollará con la última versión de OpenCV y Python 3.8 o superior.

3 Requisitos específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que los satisfaga y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos.

3.1 Funciones

1. Calcular el porcentaje del área defectuosa.

Origen: Las piezas de madera pueden poseer varias imperfecciones, el cliente pide determinar qué porcentaje de la pieza suponen dichas imperfecciones.

Para desarrollarlo y testarlo se necesitan 100 imágenes reales que contengan distintos tipos de imperfecciones.

2. Detectar imperfecciones.

Las imperfecciones podrán estar situadas en cualquier zona de la pieza además podrán disponer de distinta orientación.

Origen: Las piezas de madera no poseen las imperfecciones en la misma zona, por lo que es necesario localizar estas imperfecciones en cada pieza. Debe detectar la información independientemente de que sea un nudo o una fenda.

Para desarrollarlo y testarlo se pueden usar las imágenes usadas en el punto 1.

3. Binarizar la imagen.

Origen: Las imperfecciones poseen un tono más oscuro que el resto de la madera, para poder diferenciarlas del resto se binarizará la imagen separando el fondo del resto.

Para desarrollarlo y testarlo se pueden usar las imágenes usadas en el punto 1.

3.2 Rendimiento

- Carga de trabajo que se espera. Se procesarán cada una de las piezas de madera por separado y en solitario.
- Tasa de acierto/error que espera/acepta el cliente. Preguntar al cliente.
- Velocidad de respuesta que espera/acepta el cliente. El sistema de detección de imperfecciones en la madera debe ejecutarse en menos de 4 segundos por pieza.

3.3 Restricciones de diseño

- Necesidad de formación de los desarrolladores en los diferentes tipos de imperfecciones.
- Se va a usar Programación Funcional.
- El sistema no funcionará correctamente con imágenes donde las imperfecciones al binarizarlas se vean cortadas.
- El sistema no funcionará sobre imágenes JPG, ya que poseen una compresión con pérdida.

- Para este proyecto el objetivo es desarrollar un método que se integrará con el sistema informático de la empresa, además se hará uso del hardware que está ya dispone. Por lo tanto, en el presupuesto solo se incluye el salario del equipo de desarrollo que estaría compuesto por dos personas. Se estima un desarrollo de tres meses con un salario de 2.000€/mes por persona.

3.4 Atributos del sistema

- Fiabilidad frente a caídas del sistema. En el caso de que el parámetro de entrada sea una imagen no valida el método lo reportará como un error, indicando que dicho parámetro es incorrecto.
- Portabilidad del desarrollo a otros entornos hardware o software. El desarrollo sobre Python asegurará su portabilidad a sistemas MAC/Windows/Linux.
- Mantenibilidad del sistema con el paso del tiempo. El código desarrollado cumplirá las normas de estilo de la empresa (nomenclatura de variables y funciones, tamaño de las funciones, comentarios...).

4 Apéndice

Base de datos proporcionada por el cliente, la cual está dividida en dos conjuntos de elementos (train y test). Cada imagen con imperfecciones lleva asociado un fichero que indica donde se sitúan dichas imperfecciones.

Las entradas al sistema serán imágenes a color en formato PNG, estas serán procesadas y devolverán como salida el porcentaje del área defectuosa en cada una de las imágenes.