

## Avance 2. Análisis exploratorio de datos para el sistema de visión computacional para normalización y validación visual de anaqueles fríos

**Integrantes del equipo:**

Carlos Eduardo Ramírez Vázquez | A01795468

Cesar Bryam Rodriguez Aybar | A01795980

Wilberth Eduardo López Gómez | A01795997

**Profesora titular:** Dra. Grettel Barceló Alonso

**Proyecto integrador**

8 de febrero de 2026

# Introducción

Dentro de la metodología CRISP-ML, la fase de Preparación de los datos tiene como objetivo transformar los datos crudos del mundo real en un conjunto de representaciones adecuadas para el aprendizaje automático (Wirth & Hipp, 2000; Studer et al., 2021). En el contexto de este proyecto, los datos de entrada corresponden a imágenes capturadas por una cámara fija, por lo que la ingeniería de características no se aborda desde un enfoque tabular tradicional, sino desde los principios de la visión por computadora.

En lugar de diseñar manualmente variables estadísticas o categóricas, la ingeniería de características se centra en convertir las imágenes RGB originales en representaciones numéricas y geométricas que preserven la información espacial relevante para el problema (Szeliski, 2022). Este proceso incluye operaciones como la normalización de intensidades, la estandarización geométrica y la generación de estructuras explícitas, como lo son: puntos clave, vértices y relaciones espaciales, que permiten reducir la variabilidad irrelevante y facilitar el aprendizaje de patrones por parte de los modelos.

## 

# Descripción de los datos de entrada

Los datos de entrada son imágenes que se obtienen a partir de fotografías capturadas mediante cámaras posicionadas de manera estratégica para obtener un campo de visión de anaqueles y neveras que se quieran visualizar, lo cual otorga un conjunto de imágenes RGB como se muestran a continuación:



Imagen 1. Conjunto de imágenes de entrada.

Las cámaras empleadas son de un mismo modelo, por lo que no se obtiene una varianza en calidad de las imágenes, teniendo un conjunto con las siguientes condiciones:

* Efecto de ojo de pez presente.
* Resolución de 1920x1080.
* Pixel range de 0 - 255.

El resultado final del proyecto es producir una configuración en formato JSON que permita reproducir la rectificación siguiendo el estilo del JSON otorgado por el cliente.

# Normalización y estandarización de datos

Con el dataset de imágenes con una geometría consistente, se aplicaron procesos de normalización y estandarización con el fin de obtener una representación ideal para el modelado. Se realizó una verificación de las dimensiones en las imágenes para asegurar su compatibilidad con el pipeline, donde se analizó que conserven una resolución de 1920×1080 píxeles.

Adicionalmente, se establecieron criterios de estandarización para manejar posibles variaciones de resolución, definiendo el cambio de BGR a RGB de ser necesario y la normalización de su escala RGB de [0, 255] a [0, 1] para la generación del modelo.

# Generación de características (Feature Generation)

Se generaron varias características a partir de las imágenes procesadas de las neveras y anaqueles fríos, donde se identifican de manera manual los puntos clave a identificar, tal y como son los vértices y regiones de las puertas, que representan la estructura geométrica de los dos tipos de anaqueles, los cuales tienen en común el tipo de superficie rectangular que se puede identificar.



Imagen 2. Selección de los puntos clave de los anaqueles.

Finalmente, los puntos de cada puerta se guardan para posteriormente generar un archivo JSON con estos datos llamado “all\_annotations.json”.

# Transformaciones geométricas y espaciales

## Unfisheye

El primer requisito a cumplir es el de quitar el efecto de ojo de pez para rectificar cada una de las imágenes con el fin de una visualización más óptima, quedando de la siguiente forma:





Imagen 2. Resultado de rectificar la distorsión de ojo de pez.

El efecto se logra revertir principalmente por la función definida como “apply\_unfisheye\_and\_save”, la cual aplica los criterios establecidos para quitar este efecto y posteriormente, guardarlo para su posterior uso dentro del proceso de normalización y estandarización. Lo cual brinda un enfoque más claro al resultado esperado, aun cuando se generan espacios indeseados en la imagen resultante, es un buen elemento para el proceso de normalización y estandarización.

Es importante mencionar que estas transformaciones no se consideran únicamente como un preprocesamiento de los datos de entrada, sino que es más como una parte integral de la ingeniería de características, ya que se genera una nueva representación geométrica de los datos, lo cual tiene una interpretación diferente y reduce la variabilidad espacial entre imágenes, lo cual facilita la comparación y el análisis posterior entre estos dos pasos.

# Codificación y representación de características

## Normalización y Estandarización

Mediante la función “preprocess\_image”, se hace una comprobación y reajuste a las dimensiones de las imágenes para que sea compatible para la fase del modelado, donde se tienen tres resultados:

* Todas las imágenes están en 1920x1080. No se requiere resize.
* Hay más de una resolución o no coincide con 1920x1080.
* Se recomienda estandarizar a EXPECTED\_WH en preprocess\_image().

Con el dataset actual de 26 unidades, se obtuvo la primera salida, ahora bien, se mantendrá como artefacto persistente la imagen post-unfisheye obtenida previamente, que constituye la fuente de verdad del pipeline al momento de visualizar, validar o alimentar modelos. Por lo que almacenar tensores preprocesados no ofrece beneficios prácticos en el flujo de trabajo.



Imagen 3. Comparación entre las imágenes obtenidas en cada fase de preprocesamiento.

Asimismo, se empleó una segunda normalización para el JSON obtenido, donde cada keypoint empleado tendra una valor entre 0 y 1, lo cual es pertinente de comentar, ya que su uso será directamente para el modelo.

Para este apartado, no se aplicaron técnicas de codificación categórica como one-hot u ordinal encoding, esto porque el problema pertenece al dominio de la visión por computadora, la cual trabaja con datos espaciales continuos. Por lo que la aplicación de codificaciones de tipo categórico resultaría ser inapropiada y realmente no aportaría información relevante al proceso de modelado más del de clasificar, lo cual no es el objetivo de este proyecto.

# Técnicas no aplicadas y justificación metodológica

Hay varias técnicas de ingeniería empleadas en soluciones de problemas tabulares, las cuales no fueron aplicadas en este proyecto. Entre ellas se incluyen la discretización o binning, así como pruebas estadísticas como chi-cuadrado, ANOVA y análisis de correlación.

Estas metodologías asumen variables discretas e independientes, supuestos que no se cumplen en datos visuales espaciales. Su aplicación en imágenes podría romper la coherencia geométrica de la información y resultar contraproducente para el objetivo del proyecto.

# Selección implícita y extracción de características

La reducción de la complejidad del problema se llevó a cabo de manera implícita mediante el enfoque en regiones de interés y la transformación del espacio visual. Al concentrar el análisis en el plano del anaquel y descartar información irrelevante del entorno, tal como son aquellos espacios generados en el proceso de revertir el ojo de pez y espacios fuera de los anaqueles, lo cual reduce significativamente la dimensionalidad efectiva de los datos.

Conceptualmente, este enfoque guarda relación con técnicas clásicas de extracción de características como el análisis de componentes principales o el análisis factorial, en tanto busca representar la información original en un espacio de menor dimensión preservando las estructuras más relevantes, aunque adaptado al dominio de la visión por computadora.

# Impacto de la ingeniería de características en el modelado

La ingeniería de características aplicada tiene un impacto directo en la fase de modelado, donde se establece que el JSON generado a partir de los keypoints es esencial para la elaboración del modelo. Esto reduce el ruido espacial, mejora la estabilidad del entrenamiento y facilita el aprendizaje de patrones relevantes.

Asimismo, estas transformaciones contribuyen a una mejor capacidad para desarrollar y generar el modelo, lo cual permite una transición clara entre la fase de preparación de los datos y la fase de modelado, conforme a la metodología CRISP-ML.

# Conclusiones

En esta fase se logró transformar las imágenes originales en una representación geométrica estructurada, corrigiendo distorsión (undistort/unfisheye), aplicando homografía y generando anotaciones consistentes en formato JSON. Esto permitió reducir la variabilidad espacial entre imágenes y establecer una base estandarizada para el procesamiento automático de anaqueles fríos.

La ingeniería de características realizada permitió pasar de datos crudos (imágenes RGB) a una representación más compacta y significativa basada en vértices, puertas y relaciones geométricas, facilitando la preparación del sistema para etapas posteriores de automatización.

Como próximos pasos, se plantea automatizar la detección de vértices mediante un modelo de visión computacional, eliminando la selección manual de puntos. Asimismo, se integrará el pipeline completo para que el sistema procese cada imagen de forma autónoma y se genere el JSON final requerido por el cliente, donde la fase de preparación de los datos se alinea con la metodología CRISP-ML, asegurando coherencia metodológica y estableciendo una base sólida para el desarrollo del modelo final.

Finalmente, se realizarán pruebas de robustez ante variaciones de iluminación, inclinación y vibración para asegurar la capacidad de generalización del modelo en escenarios reales.

# 

# Referencias

* Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press
* Studer, R., Bui, T. D., Drescher, C., Hanuschkin, A., Winkler, S., Peters, S., & Müller, K. R. (2021). CRISP-ML(Q): A machine learning process model with quality assurance methodology. Machine Learning and Knowledge Extraction