IDEA / POSIBLE SOLUCIÓN: Abstracción del controlador mediante pipelines

Contexto: Hasta ahora, el controlador central del sistema tenía la responsabilidad directa de manejar la secuencia (a forma de ejemplo y como idea sintetizada):

preprocess(image)

model.infer()

postprocess(output)

Esto funcionaba, pero suponía que el controlador sabía exactamente qué hacer con la imagen y cómo tratar la salida del modelo, incluyendo cosas específicas como letterbox, reescalados, formatos de salida, etc. Es decir, el controlador sabía demasiado. Sabía incluso cómo se comportaba cada modelo, y eso atentaba directamente contra su función de "controlar" sin acoplarse a los detalles internos de cada tipo de tarea.

Nueva idea: El controlador no debería saber nada de preprocess ni de postprocess. Lo único que debería hacer es recibir una imagen y saber, en base a lo que le dice el config (vía JSONReader + ConfigSchemas), qué tipo de pipeline necesita usar.

Por ejemplo:

* Si el modelo es de tipo "detección", usar el DetectionPipeline.
* Si es de tipo "clasificación", usar el ClassificationPipeline.
* Si es de tipo "segmentación", usar el SegmentationPipeline.

Cada pipeline implementa su propia lógica:

* preprocesamiento específico,
* inferencia con el modelo,
* postprocesamiento y transformaciones.

El controlador simplemente hace:

pipeline = pipeline\_factory.get\_pipeline(model\_type)

result = pipeline.run(image, model\_wrapper)

Esto genera una arquitectura más limpia y extensible. Por ejemplo, si en el futuro se agrega un nuevo tipo de modelo (como super resolución, OCR, posestimación, etc.), sólo hay que crear un nuevo pipeline, sin tocar el controlador ni los demás pipelines.

Ventajas de esto:

* Bajo acoplamiento: el controlador no necesita saber nada de los detalles internos.
* Alta cohesión: cada pipeline se encarga de lo que debe hacer.
* Extensibilidad: se pueden agregar nuevos tipos de IA fácilmente.
* Legibilidad y testeo: cada parte se testea en aislamiento.

Conclusión: Este enfoque devuelve al controlador su rol de "orquestador" puro, sin contaminarlo con responsabilidades que no le corresponden. Es una idea que parece alinearse mejor con la filosofia del diseño original y que puede facilitar cambios o extensiones del sistema. Me había quedado en pausa por el tema de letterbox, pero ahora que eso se resolvió dentro del pre/post, vale la pena retomar esta línea.

Queda asentado como idea y posible dirección a consolidar el diseño final del sistema...