Implementación propia y validación del modelo SimpleNet para detección y localización de anomalías

1. Introducción y objetivo del trabajo

El objetivo de este trabajo es implementar desde cero y validar experimentalmente el modelo propuesto en el artículo **"SimpleNet: A Simple Network for Image Anomaly Detection and Localization" (Liu et al., 2023)**, implementando de cero la solución propuesta en el artículo.

A diferencia de un uso directo de la implementación oficial, este proyecto parte de una reconstrucción progresiva del modelo, analizando qué componentes son esenciales y verificando su comportamiento mediante experimentación. El énfasis está puesto en demostrar entendimiento conceptual y técnico del método, más que en replicar exactamente cada detalle del código original.

El trabajo se desarrolla en PyTorch y se evalúa sobre el dataset **MVTec AD**, utilizando dataloaders personalizados y una arquitectura definida explícitamente.

---

2. Contexto: detección de anomalías en visión por computador

La detección de anomalías en imágenes industriales consiste en identificar patrones visuales que no pertenecen a la distribución normal de los datos. En inspección industrial, esto suele implicar detectar defectos de fabricación en productos que, en condiciones normales, son visualmente muy similares entre sí.

Una característica clave de este problema es la asimetría de datos: existen muchos ejemplos normales y muy pocos defectuosos. Además, los defectos pueden variar considerablemente en forma, tamaño y localización, lo que dificulta su enumeración previa. Por ello, los enfoques no supervisados resultan especialmente adecuados.

En este trabajo se adopta un enfoque no supervisado, donde el modelo se entrena únicamente con imágenes normales y aprende una representación de la normalidad que permite detectar desviaciones durante la inferencia.

---

3. Idea central de SimpleNet y enfoque adoptado

La idea central de SimpleNet es reformular la detección de anomalías como un problema discriminativo en el espacio de características, evitando tanto la reconstrucción explícita de imágenes como la estimación compleja de densidades.

El modelo se apoya en cuatro ideas fundamentales:

1. Usar un backbone preentrenado como extractor de características locales.
2. Introducir un adaptador entrenable que ajuste esas características al dominio objetivo.
3. Generar anomalías sintéticas en el espacio de características, añadiendo ruido.
4. Entrenar un discriminador ligero que separe características normales de características perturbadas.

---

4. Implementación realizada

4.1 Backbone y extracción de características

Se utiliza un ResNet-50 preentrenado en ImageNet como backbone. Para la extracción de características se emplea la utilidad `create_feature_extractor` de torchvision, lo que permite obtener directamente las salidas de capas intermedias específicas.

En la implementación desarrollada:

- Se configuran las capas `layer2` y `layer3` como posibles fuentes de características.
- Todos los parámetros del backbone se congelan (`requires_grad = False`), asegurando que el entrenamiento se concentre exclusivamente en los módulos añadidos.

Esta decisión permite aislar el efecto del adaptador y del discriminador, y reduce el riesgo de sobreajuste.

4.2 Adaptador de características

El adaptador se implementa como una **convolución 1×1** que actúa sobre los mapas de características extraídos del backbone. Su función es:

- Ajustar las características al dominio industrial específico.
- Permitir que el modelo aprenda una representación más adecuada para la tarea de detección de anomalías.

En esta implementación:

- La dimensión del embedding es de 512+1024 canales (L2 y L3 respectivamente).
- Tras la convolución 1×1, las características se reordenan y se aplanan, de forma que cada posición espacial se interpreta como un vector independiente en el espacio de características.

Esto equivale a trabajar con parches de tamaño 1×1, lo que simplifica el pipeline y permite analizar el comportamiento del modelo sin introducir complejidad

adicional por el patching explícito.

4.3 Generación de anomalías sintéticas

Siguiendo el principio central de SimpleNet, las anomalías se generan añadiendo ruido gaussiano a las características adaptadas. Concretamente:

- Se parte de un embedding normal `z`.
- Se genera una versión perturbada `z' = z + \epsilon`, donde `\epsilon` es ruido gaussiano con desviación estándar `\sigma`.

En el código implementado, el valor de `\sigma` se fija explícitamente (`\sigma = 0.015`), lo que permite controlar la severidad de las anomalías sintéticas. Esta elección se inspira en valores utilizados en el código oficial y se ajusta para garantizar estabilidad durante el entrenamiento.

Un aspecto importante es que tanto las características normales como las perturbadas dependen del adaptador, lo que garantiza que los gradientes fluyan hacia él y que el adaptador y el discriminador se entrenen de forma conjunta, tal como describe el artículo original.

4.4 Discriminador

El discriminador se implementa como un MLP sencillo, compuesto por:

- Una capa lineal de 512 a 512 dimensiones.
- Una activación LeakyReLU.
- Una capa lineal final que produce un único escalar por embedding.

El discriminador recibe embeddings individuales (uno por posición espacial) y produce una puntuación que indica cuán normal o anómala es esa característica.

Se utiliza una pérdida con margen, que penaliza:

- Puntuaciones bajas para características normales.
- Puntuaciones altas para características perturbadas.

Esta formulación evita el uso de funciones de probabilidad explícitas (como sigmoides) y permite un entrenamiento estable basado en separación de márgenes.

---

5. Flujo de datos y entrenamiento

5.1 Dataloaders personalizados y MVTec AD

Para el trabajo experimental se utiliza el dataset MVTec AD, ampliamente empleado en la literatura de detección de anomalías industriales.

Se desarrollaron dataloaders personalizados que:

- Cargan imágenes normales durante el entrenamiento.
- Respetan la estructura de carpetas del dataset.
- Permiten, en fase de test, acceder tanto a imágenes anómalas como a las máscaras de ground truth.

Esto asegura un control total sobre el proceso de entrenamiento e inferencia y facilita la evaluación posterior.

5.2 Proceso de entrenamiento

El entrenamiento sigue un esquema directo y transparente:

1. Se cargan imágenes normales.
2. Se extraen características mediante el backbone congelado.
3. Las características se adaptan mediante la convolución 1×1.
4. Se generan características anómalas sintéticas añadiendo ruido.
5. Ambas se pasan por el discriminador.
6. Se calcula la pérdida con margen.
7. Se actualizan los parámetros del adaptador y del discriminador.

No se emplean fases adicionales ni preentrenamientos separados, ya que el objetivo es validar que el entrenamiento conjunto del adaptador y el discriminador es suficiente para aprender una frontera de separación efectiva entre normalidad y anomalía.

5.3 Inferencia y localización

Durante la inferencia:

- No se añade ruido a las características.
- El discriminador produce una puntuación por posición espacial.
- Estas puntuaciones se reorganizan en forma de mapa 2D.
- El mapa puede reescalarse al tamaño original de la imagen para visualizar la localización de anomalías.

Este proceso permite evaluar tanto la detección a nivel de imagen como la localización a nivel espacial.

---

6. Discusión y conclusiones

En este trabajo muestra la implementación trabajada del artículo. Las principales aportaciones del trabajo son:

- Una implementación clara y controlada del modelo.
- Un entendimiento explícito del artículo del adaptador y del discriminador.
- La validación de que la generación de anomalías en el espacio de características es suficiente para entrenar un detector eficaz.

El proyecto sienta las bases para futuras extensiones, como la incorporación de múltiples escalas, patching explícito o ajustes más finos de hiperparámetros, pero ya en su forma actual demuestra una comprensión sólida del método propuesto en SimpleNet.