




Proyecto Procesamiento Avanzado de Imágenes

Reentrenamiento Tiny-Yolo

Carlos Contreras Montanares



¿ Qué hacer?

- Reentrenar Tiny-Yolo.
- Las clases son objetos domésticos:
 - Cereal
 - Botellas
 - Tarros
 - Latas
 - Sobres
 - Envases
 - Etc.



Problema

- No hay una base de datos disponible con la cual entrenar.
- De existir, el entrenamiento se haría siguiendo los pasos establecidos en su documentación.



Herramientas ocupadas

- Python 3
- OpenCV
- Darknet



Pasos a seguir:

- *Few-shot Learning based on Context-aware Network Expansion with Artificial Training Data for Picking in Warehouse Automation (Universidad de Tokio).*
 - Segmentación de objetos.
 - Data Augmentation.
 - Generación de escenas con un contexto.
 - Data Augmentation.
 - Entrenamiento.

Segmentación

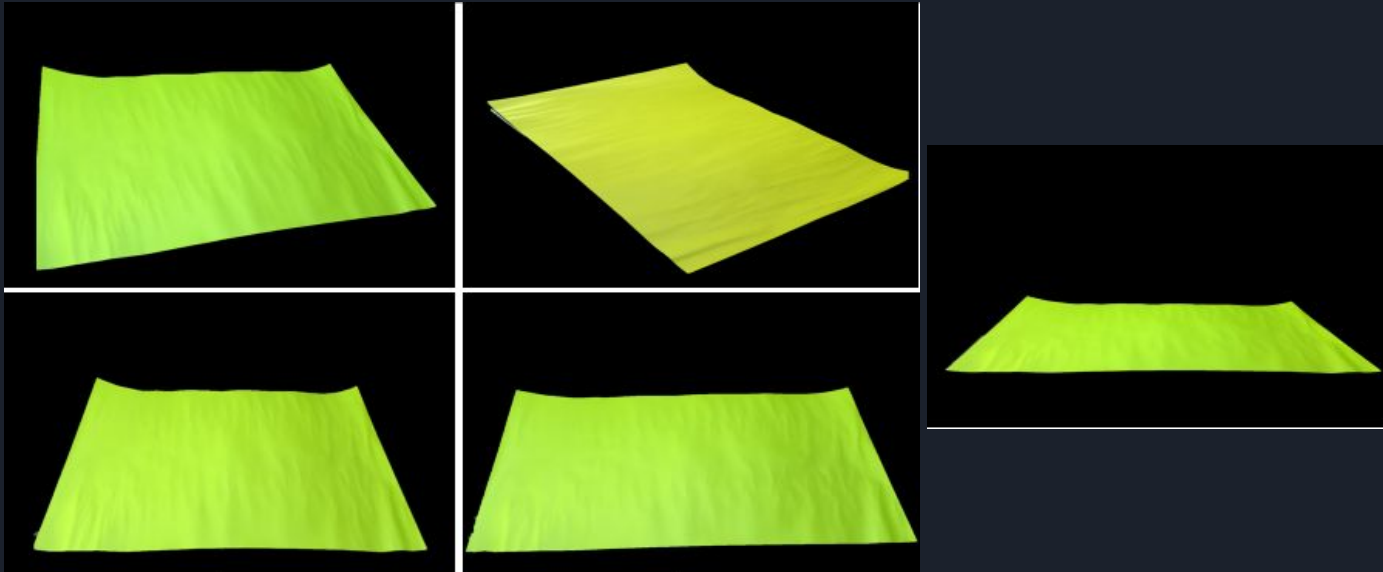


Data Augmentation

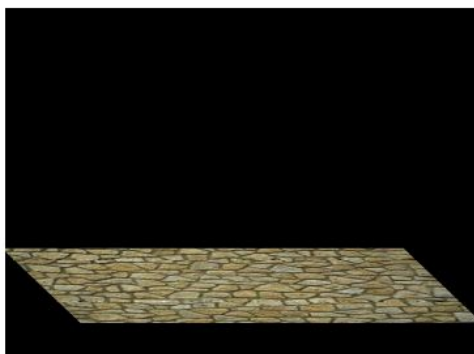
- Rotación.
- Escalamiento.
- Corte.



Escenas - Contexto:



Escena - Contexto con textura



Generación de Escenas



Data Augmentation





Entrenamiento

- GPU NVIDIA 950M 4Gb
- CPU INTEL CORE i7 INSIDE.
- 24 horas de entrenamiento.

Desempeño en conjunto de Validación

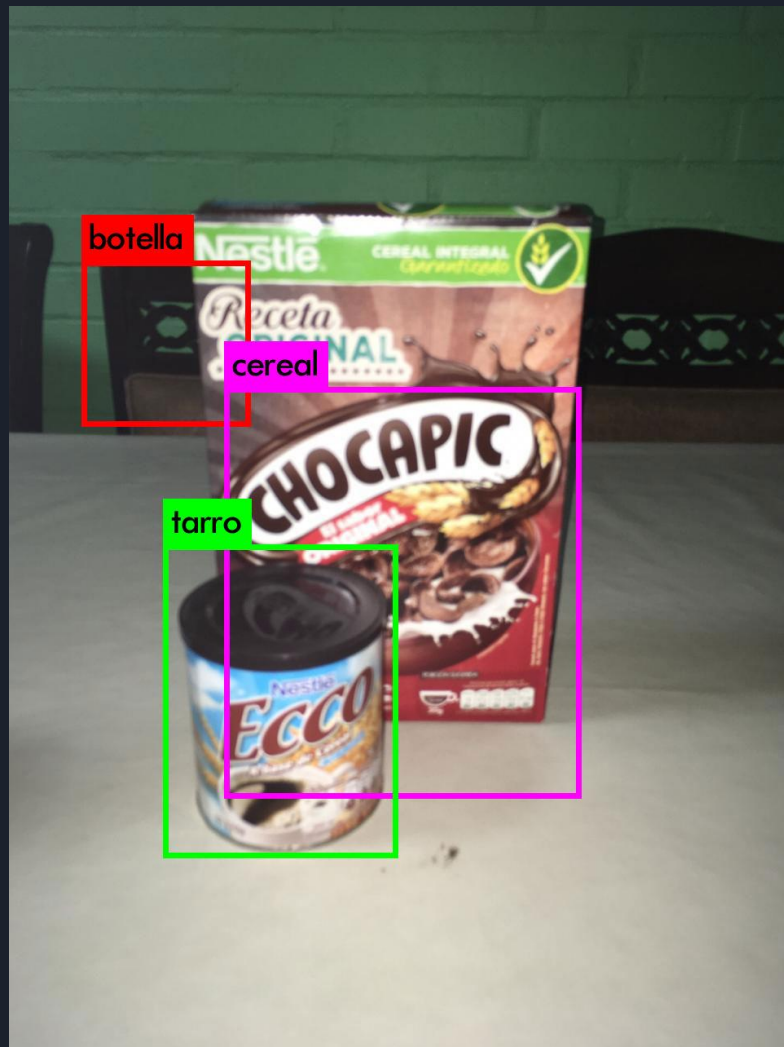
Clase	Average Precision
Cereal	90.83 %
Lata	90.84 %
Leche	90.66 %
Botella	90.72 %
Tarro	90.85 %

Average IoU = 80.50%

Recall = 92%

Detecciones:





botella

cereal

tarro



Trabajo futuro

- Probar con más elementos por clase.
- Más clases.
- Cambio de contexto por planos texturizados.
- Obtención de métricas de desempeño con un conjunto de test.



Conclusiones

- Conseguir un conjunto de entrenamiento no es trivial.
- El uso de Data Augmentation ayuda mucho a multiplicar la base que ya se tiene, pero hay que tener cuidado con que filtros ocupar.
- Hay que informarse sobre la implementación de Tiny-Yolo que se está ocupando.
- Las métricas obtenidas en la validación no necesariamente reflejan las que se obtendrían con ejemplos reales (no autogenerados).