

Refurbishing_Project_doc_withCODE

January 19, 2025

1 LEGO Bricks Identification Project: A Journey of Learning and Adaptation

1.1 1. Introduction and Motivation

Este proyecto surge como una oportunidad para aplicar y demostrar mis habilidades técnicas en el contexto de un cambio profesional hacia la ciencia de datos e inteligencia artificial. Durante esta transición, decidí combinar mi pasión por los LEGO con mi interés por la visión por computadora, dando vida a una idea que conecta creatividad y tecnología.

1.1.1 1.1 La Inspiración del Proyecto

LEGO ha sido una de mis pasiones desde la infancia. Me fascinaba cómo piezas aparentemente desordenadas podían transformarse en estructuras organizadas y funcionales. Este proyecto busca replicar esa habilidad humana, enseñando a una máquina a identificar, clasificar y organizar piezas individuales en imágenes desordenadas. Más allá de ser un simple experimento técnico, esta iniciativa refleja mi interés por explorar cómo la inteligencia artificial puede emular procesos humanos de reconocimiento visual.

1.1.2 1.2 Propósito y Objetivos

El propósito principal del proyecto es desarrollar una solución práctica y escalable que demuestre mis habilidades en ciencia de datos y visión por computadora. Los objetivos específicos incluyen:

1. Diseñar un pipeline eficiente para la detección y clasificación de piezas de LEGO.
2. Documentar el proceso de desarrollo, desde la creación del dataset hasta la implementación del modelo.
3. Generar resultados replicables y visualizaciones claras que destaquen el impacto de mi enfoque técnico.

Este proyecto también busca ser una base para aplicaciones futuras, como sistemas de clasificación robótica o asistentes visuales en tiempo real, mostrando la capacidad de escalar esta solución inicial hacia un impacto más amplio.

1.1.3 1.3 Configuración e Instalación del Entorno en Kaggle

La accesibilidad y reproducibilidad son pilares fundamentales de este proyecto. Por ello, se ha configurado un entorno basado en Kaggle Notebooks, que ofrece acceso gratuito a GPU y una integración sencilla con datasets públicos y privados.

Código de Configuración Inicial

```
[ ]: # Clone the GitHub repository
!git clone https://github.com/MiguelDiLalla/LEGO_Bricks_ML_Vision.git
%cd tu_repositorio

# Install required libraries
!pip install -r requirements.txt

# Check GPU availability
import torch
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f"Using device: {device}")
```

1.2 Justificación Técnica

1.2.1 Repositorio GitHub

Se clonará el repositorio principal del proyecto para mantener sincronizado el código y facilitar su distribución a colaboradores o empleadores.

1.2.2 Instalación de Dependencias

Las bibliotecas necesarias están especificadas en un archivo `requirements.txt`, lo que asegura que cualquier usuario pueda replicar el entorno.

1.2.3 Uso de GPU

Se verifica automáticamente si hay una GPU disponible, aprovechando las capacidades de Kaggle para entrenamiento acelerado del modelo.

1.3 Integración con Kaggle

Se recomienda subir el dataset anotado como un recurso público o privado en Kaggle. Esto permite que los notebooks puedan acceder directamente a los datos, eliminando la necesidad de cargas manuales o configuraciones adicionales.

1.4 2. Creación del Dataset

El éxito de este proyecto dependió en gran medida de un dataset bien estructurado y diverso que capturara la naturaleza de las piezas de LEGO. El proceso de creación del dataset fue dividido en dos fases principales: una inicial centrada en los ladrillos básicos y otra posterior dedicada a los studs.

2.1 Recopilación de Datos

Para la primera fase, seleccioné ladrillos básicos de mi colección, excluyendo tiles y piezas transparentes, y los clasifiqué por dimensiones. En un espacio bien iluminado, dispuse los ladrillos con suficiente separación para que fueran fácilmente distinguibles. Usando la cámara de mi móvil configurada en modo de baja calidad y captura automática, generé más de 2000 imágenes mientras caminaba lentamente por la habitación, enfocándome en ángulos típicos desde los que un usuario de LEGO observaría las piezas.

En la segunda fase, monté un set fotográfico casero utilizando una plataforma giratoria hecha con piezas de LEGO. Coloqué nuevos conjuntos de ladrillos sobre esta plataforma y capturé imágenes desde un trípode con el móvil, utilizando captura automatizada nuevamente.

2.2 Procesamiento y Normalización de Imágenes

Para garantizar la consistencia en el dataset subido a Kaggle, se desarrolló un script de procesamiento que realiza los siguientes pasos:

Redimensionamiento: Todas las imágenes fueron ajustadas a una resolución estándar para optimizar el entrenamiento del modelo.

Normalización de Nombres: Los nombres de archivo se renombraron siguiendo un formato coherente (image_#.jpg).

Código del Script de Procesamiento

```
[ ]: import os
      from PIL import Image

      # Ruta de la carpeta con las imágenes originales
      data_dir = "path_to_images"
      processed_dir = "processed_images"
      os.makedirs(processed_dir, exist_ok=True)

      # Parámetros para redimensionar
      target_size = (256, 256)

      # Procesamiento de imágenes
      for i, filename in enumerate(sorted(os.listdir(data_dir))):
          if filename.endswith(".jpg"):
              img = Image.open(os.path.join(data_dir, filename))
              img_resized = img.resize(target_size)
              new_filename = f"image_{i}.jpg"
              img_resized.save(os.path.join(processed_dir, new_filename))
              print(f"Processed {filename} -> {new_filename}")
```

1.4.1 2.1 Recopilación de Datos

Para la primera fase, seleccioné ladrillos básicos de mi colección, excluyendo tiles y piezas transparentes, y los clasifiqué por dimensiones. En un espacio bien iluminado, dispuse los ladrillos con suficiente separación para que fueran fácilmente distinguibles. Usando la cámara de mi móvil configurada en modo de baja calidad y captura automática, generé más de 2000 imágenes mientras caminaba lentamente por la habitación, enfocándome en ángulos típicos desde los que un usuario de LEGO observaría las piezas.

En la segunda fase, monté un set fotográfico casero utilizando una plataforma giratoria hecha con piezas de LEGO. Coloqué nuevos conjuntos de ladrillos sobre esta plataforma y capturé imágenes desde un trípode con el móvil, utilizando captura automatizada nuevamente.

1.4.2 2.2 Anotación de los Datos

En la primera fase, utilicé LabelMe para anotar las piezas con bounding boxes en las más de 2000 imágenes, eliminando aquellas que estaban borrosas. Posteriormente, desarrollé scripts para convertir estas anotaciones al formato compatible con YOLO y organizar las carpetas de acuerdo con los requerimientos del modelo.

En la segunda fase, utilicé un modelo entrenado inicialmente para detectar ladrillos y generé anotaciones de puntos (point annotations) dentro de las bounding boxes recortadas. Un script adicional transformó estos puntos en nuevas bounding boxes dinámicas, adaptándose a las dimensiones y posiciones de los studs detectados.

1.4.3 2.3 Limpieza y Preparación del Dataset

Las imágenes borrosas o redundantes fueron eliminadas. Para mitigar la falta de calidad en algunas capturas iniciales, implementé técnicas de aumento de datos (data augmentation) directamente en el alimentador del modelo durante las sesiones de entrenamiento. Esto incluyó rotaciones, variaciones de iluminación y espejado para incrementar la diversidad de las muestras.

1.4.4 2.4 Insights para Iteraciones Futuras

- **Automatización en la anotación inicial:** Considera herramientas como Supervisely o plataformas que integren anotación semiautomática para acelerar procesos iniciales de bounding boxes.
- **Mejorar la calidad inicial:** Aunque la baja calidad fue útil para la primera fase, para futuros proyectos se recomienda una resolución estándar para maximizar el detalle en las imágenes.
- **Balance de clases:** Implementar análisis de distribución más temprano en el proceso puede garantizar un balance adecuado en el dataset y evitar clases subrepresentadas.
- **Data augmentation externo:** Herramientas como Albumentations o Fastai podrían ofrecer técnicas más avanzadas de aumento, como distorsiones geométricas y filtros, fuera del alimentador del modelo.
- **Pruebas preliminares del modelo:** Entrenar modelos pequeños desde el inicio puede revelar rápidamente problemas en el dataset, como errores de anotación o distribuciones desbalanceadas, y ahorrar tiempo en iteraciones posteriores.

1.5 3. Selección del Modelo

1.5.1 3.1 Elección del Modelo

El desafío central del proyecto era diseñar un modelo que pudiera identificar piezas individuales de LEGO en imágenes desordenadas y con condiciones variables de iluminación y fondo. Para abordar este problema, era crucial elegir un modelo que equilibrara precisión, rapidez y facilidad de implementación. Tras investigar varias opciones, seleccioné YOLO (You Only Look Once), específicamente la versión YOLOv8n, por las siguientes razones:

- **Simplicidad y Configuración Inicial:** YOLO ofrecía una configuración amigable para mi hardware con capacidades de GPU limitadas. Su implementación directa me permitió centrarme más en la preparación del dataset y en los ajustes iterativos.
- **Velocidad en Detección en Tiempo Real:** YOLO es conocido por su capacidad de realizar detecciones en una sola pasada de red neuronal, optimizando tanto el tiempo como

el uso de recursos computacionales. Esto resultó ideal para entrenar y validar rápidamente el modelo con un dataset en constante refinamiento.

- **Robustez en Ambientes Complejos:** Su arquitectura permite manejar imágenes con múltiples objetos, incluso en escenarios donde las piezas están parcialmente ocluidas, lo que alineaba perfectamente con las condiciones del proyecto.

1.5.2 3.2 Estrategia de Entrenamiento y Adaptación

El proceso de entrenamiento incluyó los siguientes pasos:

- **Creación y Preparación del Dataset:** Se definió un pipeline para convertir anotaciones en formato LabelMe a un formato compatible con YOLO. Esto incluyó la limpieza de datos, aumento (augmentation) durante el entrenamiento y un split 80-20 para entrenamiento y validación.
- **Hiperparámetros:** Ajusté los siguientes parámetros clave para optimizar el rendimiento:
 - Tamaño de imágenes: Utilicé imágenes redimensionadas para maximizar la compatibilidad con el modelo.
 - Épocas: Comencé con 50 épocas para pruebas rápidas, aumentando gradualmente según los resultados.
 - Tasa de aprendizaje inicial: 0.001, con un decaimiento cíclico (`cos_lr`).
- **Uso de Scripts Personalizados:** Implementé scripts para:
 - Recortar piezas detectadas en imágenes, simplificando tareas posteriores de clasificación.
 - Transformar anotaciones de puntos clave en bounding boxes adaptativas según los studs detectados y dimensiones de entrada.
- **Iteraciones y Validación:** Cada iteración del modelo fue validada visualizando predicciones sobre nuevas imágenes del dataset, ajustando hiperparámetros según las métricas de precisión y recall.

1.5.3 3.3 Ejemplos y Visualizaciones

Para esta sección se incluirán:

- Imágenes con Bounding Boxes de ladrillos detectados: Visualizaciones de ejemplos representativos de detecciones exitosas, resaltando piezas con múltiples colores y tamaños.
- Comparación antes y después del modelo: Mostrar ejemplos de imágenes con y sin anotaciones generadas automáticamente por el modelo.
- Visualización de transformación de keypoints a bounding boxes: Ejemplos ilustrativos de cómo las anotaciones de studs se convirtieron en cajas 2D.

1.6 4. Próximos Pasos

El desarrollo iterativo del modelo ha proporcionado un camino claro para expandir el alcance del proyecto. A medida que se continúe mejorando, estas iteraciones ayudarán a explorar nuevas áreas de aplicación y optimización en el reconocimiento visual de LEGO.

1.7 4. Implementación del Modelo y Validación

1.7.1 4.1 Entrenamiento Inicial y Ajustes

Una vez seleccionado el modelo YOLOv8n, el entrenamiento se realizó iterativamente para ajustar el modelo a las condiciones específicas del dataset:

- **Inicio del Entrenamiento:** Se utilizó el dataset procesado y anotado para realizar pruebas iniciales con un modelo preentrenado. Estas pruebas se centraron en establecer una base para los parámetros clave, como el tamaño de las imágenes y el número de épocas.
- **Evaluación Continua:** Durante el entrenamiento, se monitorearon métricas como la precisión y el recall para identificar áreas de mejora. Las visualizaciones de las predicciones ayudaron a identificar sesgos o problemas en el dataset.

1.7.2 4.2 Validación en Escenarios Controlados

Para evaluar el rendimiento del modelo:

- **Pruebas en Dataset Anotado:** Se utilizaron imágenes del conjunto de validación para comparar la detección y clasificación con las etiquetas originales. Esto permitió identificar las clases o condiciones donde el modelo tenía mayor dificultad.
- **Pruebas con Transformaciones:** Se aplicaron aumentos artificiales (rotaciones, cambios de iluminación, etc.) para evaluar la robustez del modelo ante variaciones comunes en la captura de imágenes.

1.7.3 4.3 Resultados en Imágenes Reales

El modelo fue probado en escenarios más diversos:

- **Colección No Anotada:** Se seleccionaron imágenes reales del entorno, no incluidas en el entrenamiento, para validar la capacidad del modelo en condiciones no vistas.
- **Evaluación de Robustez:** Se analizaron las predicciones en piezas o configuraciones no consideradas en el diseño inicial del dataset, evaluando su capacidad de generalización.

1.7.4 4.4 Visualizaciones y Métricas

Para esta sección se incluirán:

- **Predicciones con Bounding Boxes:** Imágenes que muestran cómo el modelo detecta las piezas, con especial atención a las detecciones erróneas o falsas positivas.
- **Estadísticas Generales:** Tablas y gráficos que presentan las métricas clave: precisión, recall, F1-score, y tiempo promedio por predicción.
- **Análisis Comparativo:** Comparación de ejemplos con predicciones y etiquetas originales, mostrando casos representativos de éxito y áreas de mejora.
- **Visualización Tipo Spread Grid:** Una visualización tipo grid de miniaturas, mostrando imágenes anotadas aleatorias de todo el dataset.

1.8 5. Próximos Pasos

El desarrollo iterativo del modelo ha proporcionado un camino claro para expandir el alcance del proyecto. A medida que se continúe mejorando, estas iteraciones ayudarán a explorar nuevas áreas de aplicación y optimización en el reconocimiento visual de LEGO.

1.9 5. Reflexión y Futuro

1.9.1 5.1 Lecciones Aprendidas

Este proyecto no solo amplió mis conocimientos técnicos en visión por computadora y machine learning, sino que también reforzó habilidades esenciales como:

- **Gestión de Proyectos Técnicos:** Desde la conceptualización hasta la ejecución, aprendí a planificar y priorizar tareas clave para lograr un resultado completo.
- **Iteración y Resolución de Problemas:** Los desafíos encontrados, como la selección inicial de MediaPipe y ajustes en el dataset, me enseñaron a ser flexible y adaptarme rápidamente.
- **Documentación y Visualización:** Documentar el proceso de manera clara y crear visualizaciones efectivas fue crucial para comunicar el impacto del trabajo.

1.9.2 5.2 Mejoras Futuras

Si pudiera comenzar de nuevo, consideraría:

- **Ampliar el Dataset Inicial:** Incluir más clases y piezas más variadas desde el principio habría mejorado la capacidad del modelo para generalizar.
- **Explorar Alternativas a YOLO:** Experimentar con otros modelos para comparar rendimiento y adaptabilidad.
- **Automatizar Anotaciones:** Implementar técnicas más avanzadas para reducir el tiempo dedicado a tareas manuales.

1.9.3 5.3 Próximos Pasos

- **Optimización del Modelo:** Afinar los hiperparámetros y explorar técnicas de fine-tuning más avanzadas.
- **Ampliación del Dataset:** Incorporar nuevos tipos de piezas, incluidos tiles y elementos transparentes.
- **Desarrollo de una Interfaz:** Crear una aplicación interactiva que permita a los usuarios cargar imágenes y obtener predicciones en tiempo real.
- **Exploración de Nuevas Aplicaciones:** Investigar cómo este pipeline podría adaptarse a otros dominios, como la clasificación de piezas en manufactura o reciclaje.

Con estos pasos, espero seguir construyendo sobre este proyecto y expandiendo su alcance e impacto potencial.