

Tierra del Fuego, Antártida e Islas del Atlántico Sur. República Argentina.	Versión 1.0	F. Creación:  01/12/2024	Pagina 1/4	
Equipo Practicas Prof BGH	PROCEDIMIENTO USO PLACAS Ctrl			
Centro Educativo Técnico de Nivel Superior "Malvinas Argentinas"	Autor: Equipo Practicas Prof	Revisor: Equipo Practicas Prof		

Informe Técnico del Proyecto: BGH Placas Codigo

1. Introducción

El proyecto **BGH** tiene como propósito desarrollar un sistema de detección y análisis de objetos utilizando el modelo **YOLOv8**. A través de este sistema, se busca identificar componentes específicos a partir de una captura de video en tiempo real, aplicando un modelo previamente entrenado. Este informe documenta la estructura del proyecto, el flujo de trabajo, y el código implementado para la funcionalidad principal.

2. Estructura del Proyecto

El proyecto se organiza según una estructura modular basada en la metodología **Cookiecutter**, lo que garantiza un desarrollo ordenado y escalable.

2.1 Estructura del Proyecto

arduino

Copiar código

```
BGH_PowerCheck/  
├─ dataset/  
│   ├── train/  
│   └── val/  
│       └─ LevelMeToYolo/  
├─ scripts/  
│   ├── runs/  
│   │   └─ detect/  
│   ├── camera.py  
│   ├── train_yolo.py  
│   ├── validate_annotations.py  
│   └─ yolov8m.pt  
├─ docs/  
├─ notebooks/  
├─ references/  
├─ reports/  
├─ tests/  
├─ config/  
├─ utils/  
├─ bgh_powercheck.db  
├─ requirements.txt  
├─ README.md  
└─ Makefile
```

Tierra del Fuego, Antártida e Islas del Atlántico Sur. República Argentina.	Versión 1.0	F. Creación:  01/12/2024	Pagina 2/4	
Equipo Practicas Prof BGH	PROCEDIMIENTO USO PLACAS Ctrl			
Centro Educativo Técnico de Nivel Superior "Malvinas Argentinas"	Autor: Equipo Practicas Prof	Revisor: Equipo Practicas Prof		

2.2 Componentes Principales

- dataset:** Contiene los datos de entrenamiento y validación organizados en subcarpetas.
- scripts:** Incluye los scripts Python necesarios para entrenar el modelo, validar los datos y ejecutar el sistema en tiempo real.
- docs:** Documentación técnica y manuales de usuario.
- bgh\_powercheck.db:** Base de datos con configuraciones o registros.
- Makefile:** Automación de tareas comunes, como la instalación de dependencias.

3. Flujo de Trabajo del Sistema

3.1 Preparación de Datos

Conversión de anotaciones al formato compatible con YOLO utilizando LevelMeToYolo.

Validación de las anotaciones mediante el script validate\_annotations.py.

3.2 Entrenamiento del Modelo

Uso del script train\_yolo.py para entrenar el modelo con los datos de las carpetas train y val.

3.3 Implementación del Sistema

El script camera.py captura video en tiempo real, carga el modelo YOLO, y realiza inferencias para detectar componentes específicos.

3.4 Resultados

Las detecciones se almacenan en scripts/runs/detect, y los resultados se pueden visualizar en tiempo real.

4. Código Principal: camera.py

El script principal para la detección en tiempo real utiliza las siguientes características:

1 Carga del Modelo YOLO

MODEL\_PATH = "C:/Practicas\_Profesionizantes\_2\_2024/BGH\_PowerCheck/scripts/yolov8m.pt"

Tierra del Fuego, Antártida e Islas del Atlántico Sur. República Argentina.	Versión 1.0	F. Creación:  01/12/2024	Pagina 3/4	
Equipo Practicas Prof BGH	PROCEDIMIENTO USO PLACAS Ctrl			
Centro Educativo Técnico de Nivel Superior "Malvinas Argentinas"	Autor: Equipo Practicas Prof	Revisor: Equipo Practicas Prof		

2 Validación de Componentes Detectados

```
def validate_components(detections):  
  
    detected_classes = [detection['class'] for detection in detections]  
  
    detected_names = [EXPECTED_COMPONENTS[cls] for cls in detected_classes]  
  
    missing_names = [name for idx, name in EXPECTED_COMPONENTS.items() if idx not in  
detected_classes]  
  
    return detected_names, missing_names
```

3 Interfaz Gráfica con Tkinter

```
def iniciar_camara():  
  
    cam_window = tk.Toplevel()  
  
    cam_window.title("BGH Placas - Ctrl Cámara")  
  
    ...
```

4 Procesamiento de Video

```
def procesar_video():  
  
    cap = cv2.VideoCapture(0)  
  
    while not detener_camara:  
  
        ret, frame = cap.read()  
  
        detections = model.predict(frame)  
  
        ...
```

5. Tecnologías Utilizadas

- **YOLOv8**: Para la detección de objetos.
- **OpenCV**: Para el procesamiento de video en tiempo real.
- **Tkinter**: Para crear la interfaz gráfica de usuario (GUI).
- **Python 3.10+**: Lenguaje principal del proyecto.

Tierra del Fuego, Antártida e Islas del Atlántico Sur. República Argentina.	Versión 1.0	F. Creación:  01/12/2024	Pagina 4/4	
Equipo Practicas Prof BGH	PROCEDIMIENTO USO PLACAS Ctrl			
Centro Educativo Técnico de Nivel Superior "Malvinas Argentinas"	Autor: Equipo Practicas Prof	Revisor: Equipo Practicas Prof		

- 6. Resultados y Evaluación
- 6.1 Precisión del Modelo

El modelo entrenado ha mostrado alta precisión en la detección de los componentes esperados, como lo demuestra la validación visual en tiempo real.

- 6.2 Detección en Tiempo Real

El sistema es capaz de detectar los componentes especificados en el archivo COMPONENT\_NAMES con un procesamiento fluido.

- 6.3 Eficiencia de la Estructura

La estructura de **Cookiecutter** ha permitido un desarrollo ágil y ordenado.

- 
- 7. Conclusión y Recomendaciones

El proyecto **BGH** demuestra ser una solución efectiva para la detección en tiempo real de componentes.  
Se recomienda:

1. Mejorar la interfaz gráfica para incluir estadísticas de detección.
2. Ampliar el modelo para detectar más tipos de componentes.
3. Documentar cada módulo de manera más detallada para futuros desarrollos.