**INFORME DE INVESTIGACIÓN**

**Autor: Mario Zoser Mena Acha**

**Tema: Detectar y catalogar la maquinaria pesada y diversos elementos utilizados en minería de un dataset utilizando estrategias basadas en aprendizaje profundo**

1. **Introducción**
   * La visión por computadora es un campo de la inteligencia artificial y la informática que se centra en el desarrollo de sistemas que pueden adquirir, procesar, analizar y comprender imágenes del mundo real.
   * Se cuenta con un dataset de 1572 imágenes con su respectivo detalle de los elementos que lo componen a través de su respectivo archivo “.xml”
   * Este dataset es una colección de vídeos descargados de youtube contenido información de maquinaria pesada en minas a tajo abierto desde vistas aéreas.
2. **Objetivos del Proyecto:** 
   * **Detectar y catalogar la maquinaria presente en las imágenes tal como bulldozer, camión, pala, etc**
   * **Utilizar las estrategias explicadas y detalladas en clase para detectar las imágenes y clasificarlas.**
   * **Brindar información del esquema de trabajo y los pasos necesarios para lograr el objetivo.**
   * **Explicar cuantitativamente y cualitativamente los resultados del modelo de clasificación.**
3. **Metodología:** 
   * **Se utiliza la aplicación Python como herramienta de trabajo y sus respectivas librerías para obtener los objetivos del proyecto.**
4. **Contenido del Proyecto:** 
   * **Transfer Learning**
   * **Fine Tuning**
5. **Problemática**
   * ¿Qué hacer con esta información?
   * ¿Existe alguna forma de analizarla de manera individual y/o global?
   * ¿Los archivos “.xml” son congruentes?
   * ¿La información es relevante?
   * ¿podemos clasificarlos?
6. **Esquema de trabajo**
   * **El entorno de trabajo es python:**
   * **Se analiza un dataset de 1572 imágenes con su respectivo archivo de información.**
   * **Se utiliza un enfoque multiclase (09 clases).**
   * **Se adapta el código para el dataset (Leer archivos .xml, etc)**
   * **Se utiliza, re-entrena y valida el modelo fasterrcnn\_resnet50\_fpn (Transfer learning + Finetuning).**
   * **Se analiza el desempeño del modelo para diferentes épocas.**
   * **Los resultados se almacenan y se prueban con el dataset de test.**
7. **Detalle del trabajo**
   * **El entorno de trabajo es python (Google Colab).**
   * **Se copian las imágenes en un repositorio de Google drive.**
   * **Se analiza un dataset de 1572 imágenes con su respectivo archivo de información. Los datos tienen:**
     + **Se divide en 1000 imágenes de entrenamiento, validación y 512 de prueba.**
     + **Se tiene 9 clases (Persona, Camión minero, Camión, Excavadora, Bulldozer).**
     + **Se analiza el dataset, se validan las clases y se busca la mejor manera de trabajarlo.**
   * **Se selecciona el modelo pre entrenado Faster R-CNN con ResNet-50-FPN pues tiene las siguientes características:**
     + **El modelo tiene clases similares al del estudio.**
     + El modelo se basa en un algoritmo de detección de objetos que consta de dos módulos principales: un módulo de generación de propuestas (RPN) que propone regiones de interés y un módulo de red neuronal convolucional que clasifica y refina estas propuestas
     + **El modelo tiene la columna vertebral formada por ResNet-50-FPN, lo que permite** extraer de características a múltiples escalas en una red neuronal convolucional.
   * **Se modifican las clases existentes para adecuarlas al modelo pre entrenado.**
8. **Resultados:** 
   * **Se evalúa el resultado global de la ejecución del código.**
   * **El resultado "average recall (ar) @[ iou=0.50:0.95 | area= large | maxdets=100 ] = 0.826" indica que al entrenar el modelo de detección de objetos utilizando el modelo on un conjunto de datos específico, se logró un promedio de recuperación de 0.826 para objetos grandes, considerando un rango de IoU de 0.50 a 0.95 y un máximo de 100 detecciones por imagen.**

**Esto sugiere que el modelo tiene una alta capacidad para recuperar correctamente objetos grandes en las imágenes, lo cual es un indicador positivo del rendimiento del modelo en la detección de objetos de gran tamaño.**

* + **El resultado "average precision (ap) @[ iou=0.50:0.95 | area= large | maxdets=100 ] = 0.747" indica que al entrenar el modelo de detección de objetos utilizando torchvision.models.detection.fasterrcnn\_resnet50\_fpn con un conjunto de datos específico, se logró una precisión promedio de 0.747 para objetos grandes, considerando un rango de IoU de 0.50 a 0.95 y un máximo de 100 detecciones por imagen.**

**Esto sugiere que el modelo tiene una capacidad sólida para realizar detecciones precisas de objetos grandes en las imágenes, lo cual es un indicador positivo del rendimiento del modelo en la detección de objetos de gran tamaño.**

1. **Conclusiones y Recomendaciones:** 
   * La elaboración del proyecto permitió ampliar los conocimientos de visión artificial y su aplicación a un entorno industrial.
   * El mayor desafío fue el procesamiento de los datos además de validar que los datos estaban desbalanceados
   * Se encontró que se necesitaba utilizar un modelo pre entrenado y sintonizado para realizar la identificación de equipos por lo tanto se escoge el “**fasterrcnn\_resnet50\_fpn**”.
   * De los resultados para 50 épocas, tenemos que, en ciertos valores de áreas, el modelo no es eficiente, especialmente en las más pequeñas. Una de las razones es por la falta de imágenes en estas posiciones.
   * Se recomienda modificar otros parámetros del modelo y verificar su desempeño para mejorar el desempeño en elementos que no se pueden clasificar correctamente.
   * Queda pendiente realizar la evaluación por clases a través de la matriz de confusión.
2. **Fuentes**
   * Colección de imágenes: https://gitlab.com/Aleja84/Coleccion-de-imagenes
   * Código Fuente: https://towardsdatascience.com/building-your-own-object-detector-pytorch-vs-tensorflow-and-how-to-even-get-started-1d314691d4ae
   * https://debuggercafe.com/object-detection-using-pytorch-faster-rcnn-resnet50-fpn-v2/
   * https://arxiv.org/abs/1506.01497