Ammunition Component Classification Using Deep Learning

Team:

- Akash S – 21011101013 AI&DS A
- Akash R – 21011101012 AI&DS A

Reference Paper: https://paperswithcode.com/paper/ammunition-component-classification-using

---

Introduction

Ammunition Component Classification Using Deep Learning is an innovative and crucial task aimed at revolutionizing the field of ammunition identification and sorting. With the increasing complexity of ammunition types and components, traditional classification methods are becoming less effective. In this context, deep learning techniques offer a promising solution to automatically and accurately classify various ammunition components, such as casings, bullets, and primers, from images or sensor data. By harnessing the power of deep neural networks, this cutting-edge approach holds great potential for enhancing safety, efficiency, and precision in ammunition handling, logistics, and inventory management.

Scope

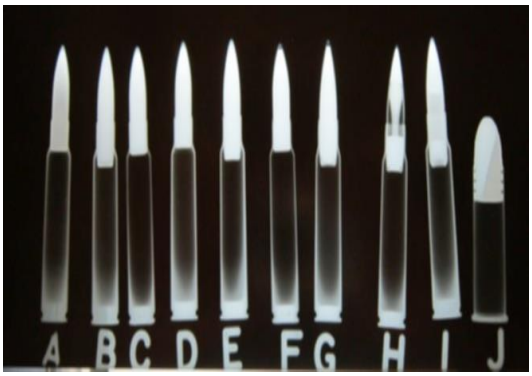
The scope for Ammunition Component Classification Using Deep Learning is vast and encompasses multiple domains within the ammunition industry. One primary area of application is in military and law enforcement settings, where accurate and rapid identification of ammunition components is critical for ensuring proper handling and usage. Furthermore, the task holds significant potential in ammunition manufacturing, streamlining the quality control process by automatically detecting faulty or mislabelled components. Additionally, the scope extends to civilian sectors, such as shooting ranges and

firearm enthusiasts, where efficient classification of ammunition components can enhance safety and optimize inventory management.

Dataset Description

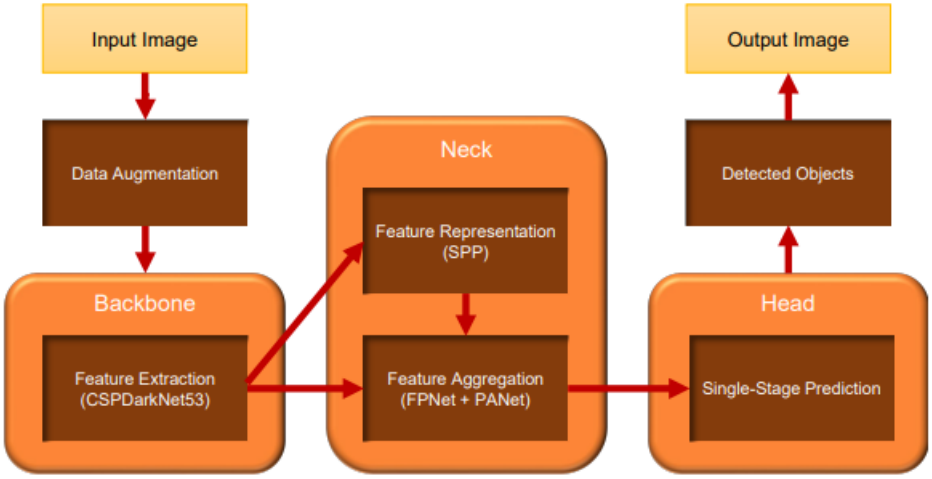
Two training datasets have been manually created by the author from visual and x-ray images of ammo. The x-ray dataset is augmented using the spatial transforms of histogram equalization, averaging, sharpening, power law, and Gaussian blurring in order to compensate for the lack of sufficient training data.

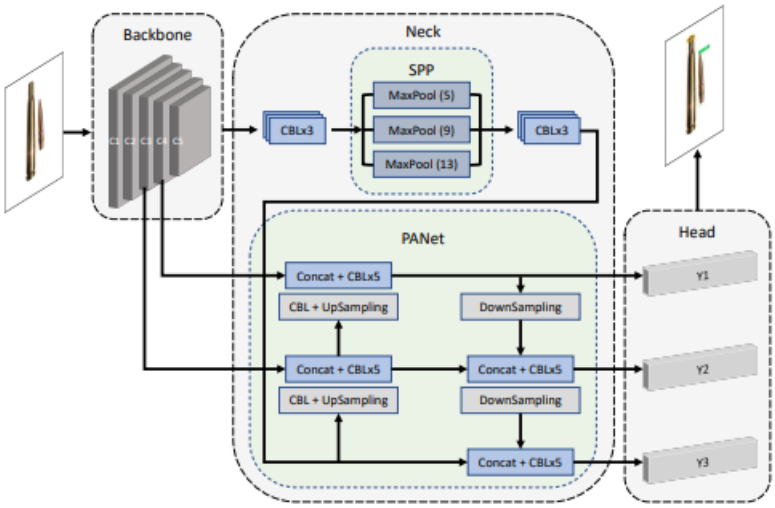
The goal is to sort the pure metal scrap from the scrap pieces that contain traces of explosive hazards. The two classes are named MDAS (Material Documented as Safe) and MPPEH (Material Potentially Possessing Explosive Hazard), respectively. Due to the lack of a sufficient number of x-ray images, several data augmentation techniques are applied as a pre-processing step.



Link to data source: https://github.com/hadign20/Scrap-Classification

Base Architecture Diagram





The YOLOv4 model has been used. The YOLOv4 architecture consists of three distinct components, namely, the backbone, the neck, and the head.

The backbone network for feature extraction is the CSPDarknet53 which is used for splitting the current layer into two parts.

The neck is the intermediate section between the backbone and the head and contains modified versions of the path aggregation network (PANet) and spatial attention module with the purpose of having a higher accuracy by information aggregation.

The head of the architecture represents the dense prediction block, which is used to locate the bounding boxes and final classification.

To enhance the model's generalization capability, YOLOv4 employs data augmentation techniques like random scaling, translation, and flipping during training.

YOLOv4 also integrates the PANet module, which stands for Path Aggregation Network. PANet helps improve information flow between different feature pyramid levels, enhancing object detection accuracy.

The Spatial Pyramid Pooling (SPP) module is used in YOLOv4 to capture multi-scale contextual information from the input image, enabling the model to detect objects at various scales efficiently.

Result Analysis

Results for both the datasets have been recorded.

Table 2. The quantitative results of testing tiny-YOLOv4 on the VACD data

	TP	FP	FN	Precision	Recall	F1-score
Full-ammo	159	50	21	76.08%	88.33%	81.75%
Casing	92	104	10	46.94%	90.2%	61.74%
Projectile	84	25	9	77.06%	90.32%	83.17%
Total	335	179	40	65%	89%	75%

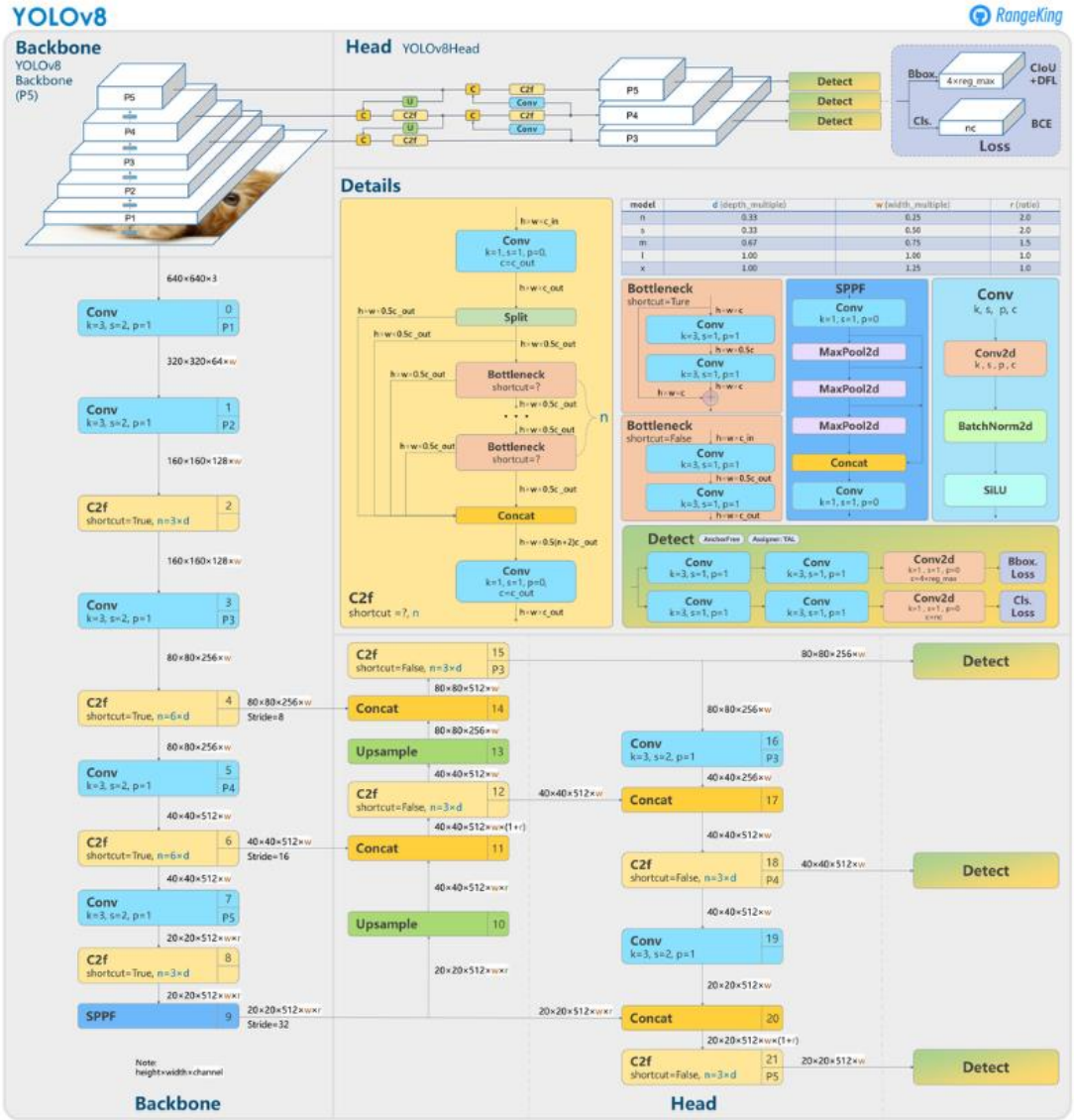
Table 3. The quantitative results of testing tiny-YOLOv4 on the XACD data

	TP	FP	FN	Precision	Recall	F1-score
Full-ammo	499	21	1	95.96%	99.8%	97.84%
Casing	41	6	7	87.23%	85.42%	86.32%
Projectile	18	0	4	100%	81.82%	90%
Total	558	27	12	95%	98%	97%

The F1-Score has been reported as 75% for the RGB data and 97% for the X-Ray data. So, there is a **scope for improvement** in the classification of RGB image data.

Proposed Architecture

The YOLOv8 Model



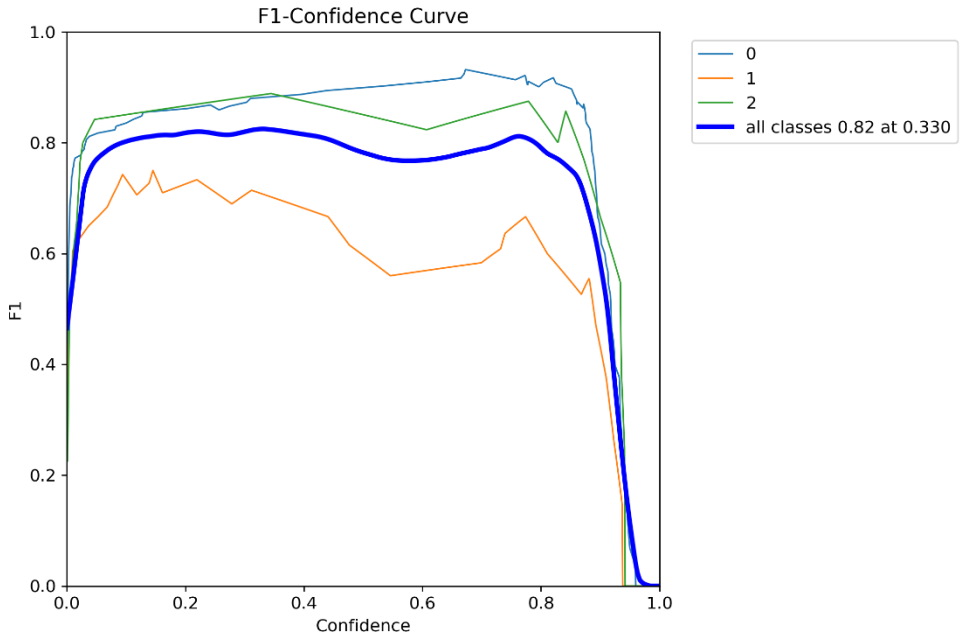
YOLOv8 has outperformed YOLOv5 in terms of accuracy. The YOLOv8s model has achieved an average precision of 51.4% on the COCO dataset, while the

YOLOv8m model has achieved an average precision of 54.2% on the same dataset. YOLOv8 has also shown superior performance in detecting small objects and has addressed some of the limitations of YOLOv5. YOLOv8 has a lower FPS than YOLOv5 on the CPU, but it still has a decent FPS for real-time applications and more FPS than YOLOv5 on some GPUs.

Improved results:

After training YOLOv8 model, the results were significantly increased than the previous. Achieved a Mean Average Precision(mAp) around 90 percent and Max F1 score of 82% at 30 percent Confidence threshold.

```
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOP
                 Class     Images  Instances      Box(P          R      mAP50
                   all         33         77      0.754      0.911      0.913
                     0         33         56      0.801      0.982      0.971
                     1         33         13      0.661      0.751      0.813
                     2         33          8      0.799          1      0.955
Speed: 0.2ms preprocess, 5.6ms inference, 0.0ms loss, 3.0ms postprocess per ima
Results saved to runs/detect/train4
ultralytics.utils.metrics.DetMetrics object with attributes:
```



Here is a Sample Prediction:

Thus, using this improved model on Visual RGB dataset eliminates the use of using X-ray input images and reducing resource usage.

Colab Link: https://colab.research.google.com/drive/109OEjGT0Aw5-nVPQd5RAfg5ETDrNj7B6?usp=sharing

Dataset : AkashS0510/Ammunation_Component_Classification (github.com)

Model weights : Ammunation_Component_Classification/train4/weights at main · AkashS0510/Ammunation_Component_Classification (github.com)