


Military Badge Detection and Classification Algorithm for Automatic Processing of Documents

Charith Gunasekara¹^a, Yash Matharu²^b and Rohan Ben Joseph³^c

¹*Department of National Defence, Government of Canada, Ottawa, ON, Canada*

²*Faculty of Engineering, McMaster University, Hamilton, ON, Canada*

³*Department of Computing Science, Simon Fraser University, Burnaby, BC, Canada*

Keywords: Computer Vision, Object Detection, Document Classification, YOLOv5.

Abstract: This paper outlines a robust approach to automate the detection of military badges on official government documents utilizing YOLOv5 computer vision model. In an era where the rapid classification and management of sensitive documents is paramount, developing a system capable of accurately identifying and classifying distinct badge types plays a crucial role in supporting data management and security protocols. To address the challenges posed by the lack of accessible, real-world government and military documents for research, we introduced a novel method to simulate training data. We employ a technique that automates the data labelling process, facilitating the generation of a comprehensive and versatile dataset while eliminating the risk of compromising sensitive information. Through careful model training and hyper-parameter tuning, the YOLOv5 model demonstrated exemplary performance, successfully detecting a wide spectrum of badge types across various documents.


1 INTRODUCTION


The sheer volume of documents generated presents a unique challenge in large-scale organizations such as governmental departments and the military. Traditional manual methods of document classification are not only labour-intensive but also time-consuming. In the pursuit of efficiency, there's a growing trend towards automation. However, this path isn't devoid of challenges. For one, access to open data for research is restricted, often due to strict organizational security policies. This limits the potential to utilize vast internal datasets for training sophisticated machine learning models (Brown, 2010). Such constraints are indeed a missed opportunity, especially when machine learning algorithms have showcased proficiency in tasks demanding domain knowledge and uncompromising attention (Orosz et al., 2022).


The appeal of document classification through machine learning is evident, yet earlier attempts in this direction often stumbled due to data scarcity and the tedious nature of data labelling (Song et al., 2019)

(Ciecierski and Kamola, 2020) (Huber-fliflet et al., 2019). In a novel approach, (Chiu et al., 2010) leveraged Optical Character Recognition (OCR) for extracting textual content and employed the Normalized Cuts algorithm for clustering non-textual pixels. Although innovative, this methodology was heavily reliant on manually labelled data, reducing its scalability.

A unique approach was introduced by (Kallempudi et al., 2022), who proposed the "Soft Teacher" mechanism. This semi-supervised pipeline catered to graphical object detection within scanned document images, even when working with limited labelled data. Similarly, (Arvind, 2023) integrated OCR for keyword vector selection in classifying government documents. Yet, their endeavours were limited by training data, a recurring issue predominantly due to access restrictions. Delving deeper into automating document feature recognition, (Forczmański et al., 2020) showcased a technique that employed Convolutional Neural Networks (CNN) to automatically segment various elements, such as logos, stamps, and text blocks, from paper documents. Their CNN-centric approach was validated as superior when compared with the conventional cascade-based detection method.

^a <https://orcid.org/0000-0002-7213-883X>

^b <https://orcid.org/0009-0003-8635-4239>

^c <https://orcid.org/0000-0001-8069-5874>

A significant paradigm shift in the realm of object detection was heralded by the birth of the You Only Look Once (YOLO) algorithm (Redmon et al., 2016). YOLO, and its subsequent iterations showed the first one-stage detection mechanism in the deep learning, outclassing many existing algorithms in accuracy and speed (Zou et al., 2023). Though alternative one-stage object detection models like RestNet have been explored, especially for tasks like logo detection (Sarwo et al., 2019), YOLO is still leading in terms of efficiency (Deng et al., 2023). Building on YOLO's foundation, (Bailey et al., 2022) crafted a training dataset for bounding boxes tailored for varied object detection. Meanwhile, (Rezkiani et al., 2022) employed YOLOv4 to discern logos on university diplomas, aiming for document classification. However, the prevailing challenge remains the manual labelling of objects and bounding boxes, which inevitably makes the entire process labour-intensive. Several versions of YOLO have been introduced since its inception. Table 1 showcases the popularity of each version through their respective Github stars. While the newer version YOLOv8's real advantage comes in applications that require real-time object detection, YOLOv5 is preferred for still object detection primarily due to its robust Pytorch-based ecosystem and extensive documentation support.

While advancements have been made in object detection and document classification, a real gap exists regarding specialized domains like military and governmental sectors. The lack of publicly available data due to organizational security policies and manual labour involved in creating training datasets in the labelling and marking bounding boxes severely throttles the speed and efficiency of the model training process. When considering the volume of official documents generated daily, this poses a significant bottleneck. We introduce an algorithm capable of autonomously creating training datasets, eliminating the requirement for manual labelling of data. By employing data augmentation techniques, our algorithm generates synthetic dataset to replicate the features and complexities of real-life scanned sensitive documents. This not only avoids labour-intensive manual labelling but also paves the way for scalable and efficient document classification in this domain.

2 RESEARCH METHODOLOGY

2.1 Data Collection

The overarching objective of this research is to classify Canadian military documents based on the

Table 1: GitHub Popularity of YOLO Versions.

YOLO Version	GitHub Stars in 1000s
YOLO v3	9.7
YOLO v4	21
YOLO v5	41.9
YOLO v6	5.2
YOLO v7	11.3
YOLO v8	13.2

badges imprinted on them. To achieve this, a comprehensive dataset was required to train our model efficiently. Utilizing existing documents with badges for this purpose was not deemed practical due to two primary concerns:

a) Training an image detection model effectively mandates thousands of training samples for each class. Manual labelling of such vast quantities of data proves to be labour-intensive and time-consuming. b) The pre-existing documents house sensitive information, making them unsuitable for open research.

Given these limitations, the decision was made to develop a mock dataset mirroring the patterns found in these military documents. The process for constructing this mock dataset is detailed below:

1. **Badge Compilation.** An exhaustive collection was undertaken of badges from various Canadian military organizations to serve as our primary dataset.
2. **Document Template Collection.** We procured internal document templates and unfilled forms, ensuring that they do not contain any sensitive information. This provided us with the base structure over which badges could be overlaid.
3. **Mock Document Creation.** A simulated set of labelled documents was constructed using the gathered badges and document templates. The methodology adopted for pasting the badges onto the documents involves a data augmentation algorithm, elaborated in the subsequent section.

The abovementioned approach ensured the synthesis of a robust dataset, eliminating risks associated with using genuine documents and significantly reducing manual labelling efforts. This mock dataset, we believe, will sufficiently represent the sophistication of real-world military documents, aiding in the effective training of our image detection model.

2.2 Data Pre-Processing and Augmentation

In order to provide an effective training ground for the YOLO5 model, we undertook extensive data pre-processing and augmentation measures. Our dataset combined both the badge and document datasets to

Table 2: Badge and Logo Data Sources and URLs.

Data Source	URL
Gallery of Canadian Force Badges	https://www.canada.ca/en/services/defence/caf/military-identity-system/canadian-forces-badges.html
Government of Canada Logo	https://www.international.gc.ca/world-monde/assets/images/funding-financement/canada-aid-aide/canada-wordmark-colour.jpg
National Defence Logo	https://media.socastsr.com/wordpress/wp-content/blogs.dir/1977/files/2021/05/national-defence.png
DRDC Logo	https://www.canada.ca/content/dam/drdc-rddc/images/articles/2021/drdc-logo.jpg

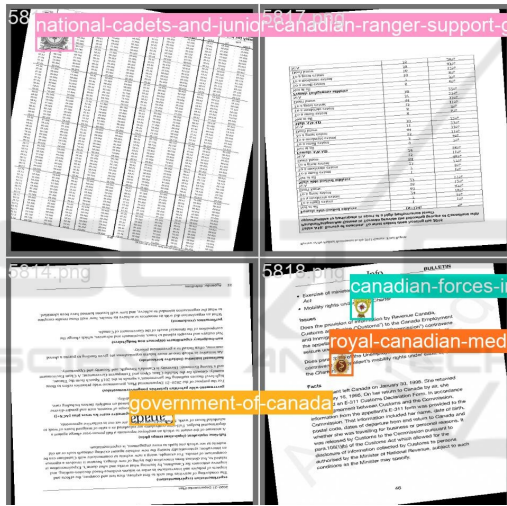


Figure 1: Sample dataset of generated images.

create an encompassing platform for model training. Adhering to best practices as suggested by the YOLO V5 model’s library, we established a target of at least 1,100 images per class. This culminated in a training dataset with 55,400 labelled objects per class and a subsequent validation set comprising 13,860 images distributed evenly across all badge and logo classes.

We ensured the inclusion of images depicting varied environmental conditions – from assorted lighting scenarios to differing viewing angles and diverse image sources like online scrapes, local collections, and captures from various camera types.

To effectively train the model to adapt to real-world conditions, each image underwent rigorous processing:

1. Image Resizing. All images were resized to 640 x 640 pixels, the default resolution supported by the

YOLOv5 model.

2. Basic Augmentations. These included random horizontal and vertical flips with a 50% probability, along with random rotations in the range of -15 to 15 degrees.
3. Badge Integration and Augmentation. Randomly selected documents were resized to the default YOLO V5 resolution. Then, anywhere from 0 to 10 badges were randomly selected and underwent the aforementioned augmentations. Badge sizes were varied (60 to 125 pixels width), with heights adjusted to keep aspect ratios intact. Badge placement was randomized within set coordinate limits, ensuring varied placement within the document, thereby exposing the model to a plethora of sizes and placements.
4. Global Augmentations. The application of global augmentations plays a crucial role in training models, particularly in creating scenarios that mimic real-world conditions. One common approach to enhance the robustness of a model is to introduce Gaussian noise during the data augmentation stage. This method helps in simulating variations and imperfections in real-world data. During the selection of augmentations, a balance between noise and accuracy was a critical consideration.

Moderate introduction of Gaussian noise can serve to regularize the model during training, thereby improving its generalization capabilities. It allows the model to become resilient to noise and slight variations in the input data, potentially reducing overfitting and enhancing accuracy when evaluating the model with real-world images that naturally contain some noise. On the contrary, a high level of Gaussian noise might degrade the model’s performance, making it difficult for the model to learn meaningful patterns from the data and possibly leading to increased error rates.

- Gaussian Noise. Gaussian noise, with a sigma (standard deviation) value of 0.5, was introduced to mimic the common noise encountered during the scanning of documents. A sigma value of 0.5 was chosen to balance the introduction of noise for robustness while maintaining the accuracy of object detection in the YOLO model. This moderate level of noise ensures that while the data contains variations, it does not become unrecognizable. Thus, images with moderate noise maintain semantic similarity to the original, allowing the model to learn robust features while managing real-world variations

(Liu et al., 2018),(Li and Ghosal, 2014).

- **Gaussian Blur.** Gaussian Blur was also utilized to account for potential losses in detail or softening of edges that can occur during scanning. That was implemented with a sigma value ranging from 0.1 to 1 and used a kernel size of (3, 3). This approach ensures that the model can adapt to potential variations in image sharpness, thereby enhancing its performance with varying image qualities (Suto, 2023). In Gaussian blurring, the kernel size influences the extent of smoothing applied to the image, making it an essential hyperparameter for tailoring image processing tasks, balancing the ability of the model to learn to interpret noisy and blurred images against the preservation of important features for the feature detectors to detect patterns from hierarchically. The kernel size of (3, 3) in Gaussian blur operations signifies a relatively small kernel, resulting in localized blurring and preserving finer details in the image. It denotes a 3x3 grid used for convolution, where each pixel's value is recalculated as an average of the neighbouring nine pixels, giving a smoothing effect.

Maintaining consistency and accuracy in labelling is crucial. Every image's labelling was ensured to be thorough and accurate, as partial labelling hampers effective training. Background images without objects were also incorporated into the dataset (representing 0-10% of the total) to reduce false positives. Each processed image was saved in PNG (Portable Network Graphics) format, signifying the culmination of both document and badge integration. Alongside, a text file was generated to serve the YOLO model, detailing badge locations in the format:

```
class_number, x_center, y_center, width, height
```

All coordinate values were adjusted to the image's dimensions for normalization purposes by dividing by the 640-pixel width or height.

The program used for the dataset generation process is explained in Algorithm 1.

2.3 Model Training

The essence of achieving optimal performance in deep learning applications lies in the careful selection and fine-tuning of hyperparameters, determining how a model learns and adapts to given data. Our focus in this section is to elucidate the training process for our chosen YOLOv5 model.

Given our dataset's size, we deemed it advantageous to initiate our training using the pre-trained

Data: Total number of pages, dataset name

Result: Pages and text files saved to output directory

initialization, Create maps for class number to badge, and document number to document filename;

for each page created do

Select random number of badges;

Select random document;

Load in document, resize to 640x640 and augment by horizontally/vertically flipping, and rotating;

Open a text file;

for each badge do

Load in badge and augment by horizontally/vertically flipping and rotating;

Paste badge onto document;

Write to txt file class and coordinates;

end

Augment combined document + badge page using gaussian noise and blur Save page as PNG file

end

Algorithm 1: Algorithm for generating the dataset.

weights from YOLOv5's default set. Such a decision often aids in faster convergence, especially when data volume is limited.

2.3.1 Hyper-Parameters

The summary of our chosen hyper-parameters and their respective roles in the training process is given below.

- **Learning Rate.** Our experiments were done using a learning rate of 0.02. The significance of this hyper-parameter lies in the step size for updating model parameters. Careful tuning is imperative to avoid rapid divergence or protracted convergence.
- **Batch Size.** Ensuring efficient computation while retaining model generalization, our batch size was set at 16. This balances between computational needs and the potential benefits of more frequent model updates.
- **Epochs.** The number of epochs for the training cycles was set to 55 to allow enough time for the model to converge the loss functions to a minimum value. At the same time, we monitored the model performance with a validation set to ensure it does not overfit.
- **Optimizer Selection.** Our model was trained using the Stochastic Gradient Descent (SGD) optimizer, chosen for its synergy with Pytorch. SGD

is an iterative process where, at each step, a subset of the training data (mini-batch) is used to compute gradients and update model parameters. Accompanying hyperparameters, such as momentum (set at 0.937) and weight decay (set at 0.0005), were carefully selected to influence the optimizer’s behaviour, ensuring efficient convergence.

- **Architectural Regularizations.** The dropout rate, fixed at 0.5, ensured periodic deactivation of neurons, enabling model robustness. Meanwhile, batch normalization, with momentum set at 0.937, contributed to training stability.

2.4 Training Environment

The model training was conducted on multiple GPUs on a cloud-based virtual machine (VM) with the following specifications:

- RAM (Random Access Memory): 224GB
- CPUs (Central Processing Units): 12 virtual CPUs (vCPUs)
- Generation: V2
- Architecture: 64-bit (x64)
- Operating System: Linux
- VRAM (Video RAM) per GPU (Graphics Processing Units) : 24GB
- Number of GPUs: 2
- Temp storage (SSD(Solid State Drives)): 1474GB
- GPU memory: 32GB
- Max data disks: 24
- Max uncached disk throughput: 40000/400 (IOPS(Input/Output Operations Per Second)/MBps(Megabytes Per Second))
- Max NICs (Network Interface Cards): 8

3 RESULTS

While training our models, we evaluated the model’s performance on a separate validation dataset, which comprised document objects unfamiliar to the models from their training phase. The subsequent subsections provide an in-depth analysis of the performance metrics obtained from this testing process.

3.1 Precision and Recall

Precision and recall are crucial indicators of a machine learning model’s performance in classification

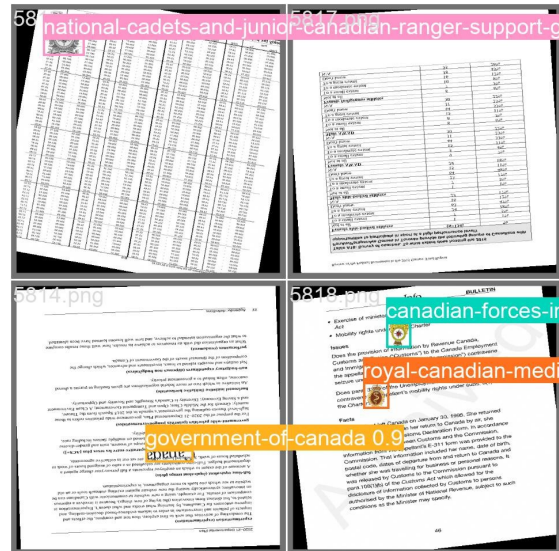


Figure 2: Sample dataset of predicted images.

tasks. While precision (Equation 1) gives us an insight into the correctness of our model by measuring the ratio of true positive (TP) predictions to the sum of true positive (TP) and false positive (FP) predictions, recall (Equation 2) on the other hand, highlights the model’s ability to identify all relevant instances by evaluating the ratio of true positive(TP) predictions to the sum of true positive (TP) and false negative (FN) predictions.

Through the graphical representations depicted in Figure 3 for Precision vs. Epochs and Figure 4 for Recall vs. Epochs, we can draw some insightful conclusions about the model’s behaviour over training epochs. As seen in Figure 3, the precision values are rising as the training epochs increase. Remarkably, after the 10th epoch, the precision converges to an impressive 99.5%. This connotes that our model, by this stage, has honed its ability to predict true positives with negligible false positive rates.

On the other hand, the recall values, as shown in Figure 4, do demonstrate improvement as the training progresses by converging to a score of 99.78%. This suggests that while the model has become adept at correctly classifying the positive cases, it still misses out on some, leading to a higher number of false negatives as compared to false positives.

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

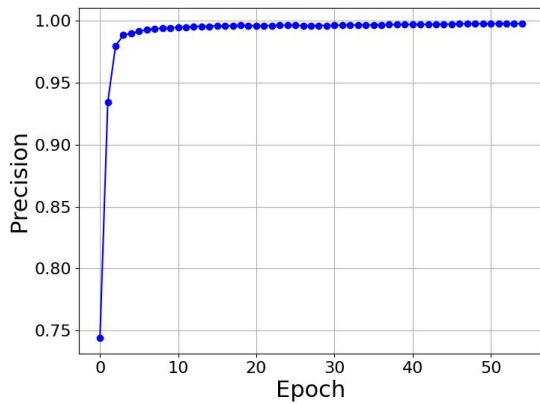


Figure 3: Precision vs Epochs Curve.

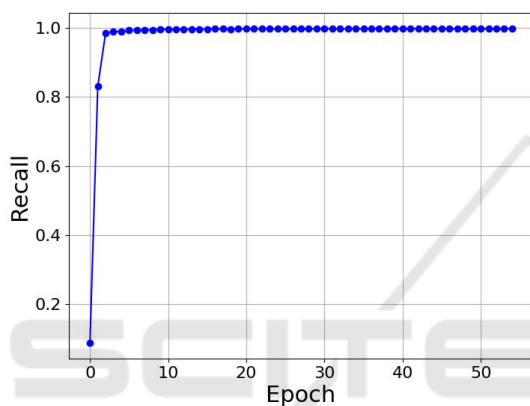


Figure 4: Recall vs Epochs Curve.

3.2 Intersection over Union (IoU)

Intersection over Union (IoU) The Intersection over Union (IoU) provides a quantitative assessment of the precision of predicted bounding boxes by measuring their overlap with respective ground truth boxes. IoU is obtained by computing the ratio of the area of intersection to the area of union between the predicted and true boxes.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

IoU values span from 0, representing no overlap, to 1, indicative of a perfect match between the predicted and actual bounding boxes. Employing IoU as a threshold metric, accurate predictions are acknowledged when the IoU surpasses a predetermined value.

As elaborated in section 3.3, the performance of the model is evaluated using a threshold of 0.5, ensuring at least 50% overlap of the predicted box with the ground truth, and a spectrum of IoU thresholds from 0.5 to 0.95, inclusive of several intermediate values. The average precision is calculated distinctly for each threshold, followed by the computation of the mean.

The mAP 0.5:0.95 furnishes a holistic evaluation of the model’s performance, encapsulating its efficacy across many overlap scenarios. This is perceived as a rigorous evaluation metric due to its imperative for high accuracy across diverse IoU levels.

3.3 Mean Average Precision (mAP)

In object detection and classification, Mean Average Precision (mAP) is a vital metric aggregating the model’s performance across different confidence thresholds. It combines precision and recall effectively, providing a holistic view of the model’s ability to identify objects and minimize false detections correctly. The mAP score is calculated by taking the average of the Average Precision (AP) values for each class or category of objects. AP, in turn, is determined by plotting the precision-recall curve for a specific class and calculating the area under that curve. (Henderson and Ferrari, 2017)

To better visualize our model’s performance throughout training, we present the mAP curves in Figures 5 and 6, illustrating how mAP values evolve across training epochs. This graphical representation aids in understanding the consistency and reliability of our model in detecting document objects throughout its training process.

The mAP’s progression over training epochs is a strong indicator of the model’s learning trajectory. A rising curve denotes a consistent enhancement in object detection capabilities during the training process. A mAP value inching closer to 1 indicates a commendable performance, where the model adeptly identifies badge regions while effectively managing false positives and false negatives.

Mean Average Precision at Intersection over Union 0.5 (mAP 0.5) (Figures 5): This metric provides an assessment of average precision, stipulating that predictions are deemed accurate when their Intersection over Union (IoU) with the corresponding ground truth bounding boxes is 0.5 or above. Essentially, it gauges the model’s proficiency in object detection where there is a moderate overlap with the ground truth annotations.

Mean Average Precision at Intersection over Union 0.5:0.95 (mAP 0.5:0.95) (Figure 6): This evaluative metric expands the assessment to encompass an array of IoU thresholds, specifically from 0.5 to 0.95, including intermediate values. It independently determines the average precision for each threshold, subsequently computing the mean of these values. The mAP 0.5:0.95 thoroughly appraises the model’s capabilities, considering its performance across an extensive range of overlap scenarios. This metric is

often viewed as a more rigorous evaluative gauge, which mandates elevated precision across assorted IoU levels. Both mAP 0.5 and mAP 0.5:0.95 serve as instrumental metrics in applied settings. The mAP 0.5, affords insights into the model’s performance under relatively forgiving conditions. In contrast, mAP 0.5:0.95 provides a more rigorous assessment, ensuring the model sustains high precision, even when object boundaries are meticulously aligned with the ground truth annotations.

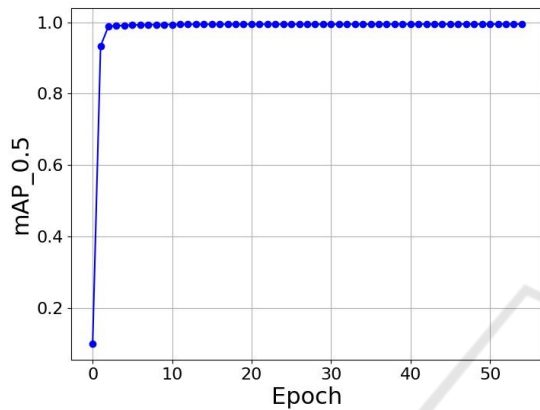


Figure 5: Mean Average Precision Curve over an IoU threshold of 0.5.

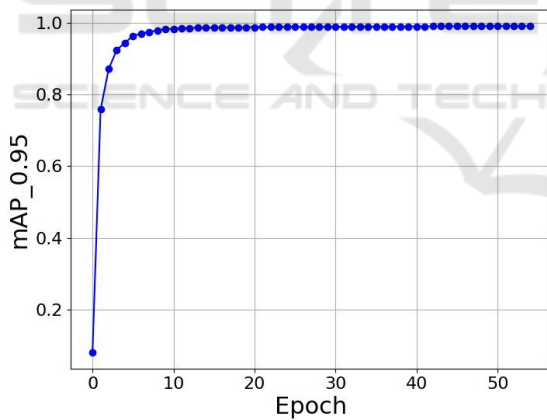


Figure 6: Mean Average Precision Curve over an IoU threshold range of 0.5 to 0.95.

3.4 F1 Scores

The F1 score, depicted in Equation 4, offers a combined evaluation of a model’s precision and recall. This metric becomes particularly insightful when classes are distributed unevenly or when the implications of false positives differ markedly from false negatives. Conceptually, the F1 score encapsulates the overlap between a model’s predictions and the ground truth. As visualized in Figure 7, we trace the

trajectory of the model’s F1 score across 55 epochs. By calculating it as the harmonic mean between precision and recall, the F1 score is a comprehensive metric, effectively harmonizing the balance between these pivotal performance indicators. As the F1 score converges to one, the model’s predictions increasingly align with the actual data, showcasing optimal precision and recall. This convergence to one indicates near-perfect harmony between detected and actual document objects, reflecting the model’s exemplary performance.

$$F1\ Score = \frac{2}{\left(\frac{1}{Precision} + \frac{1}{Recall}\right)} \quad (4)$$

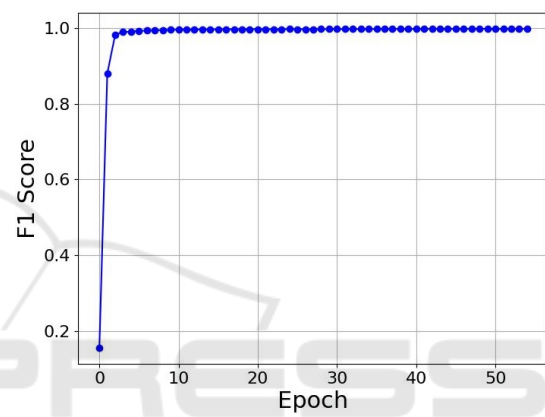


Figure 7: F1score vs Epochs Curve.

3.5 Loss Functions

Loss functions are used to quantify the difference between predicted values (output of a model) and actual target values (ground truth) during training. The three primary loss components are box loss (minimizing discrepancies in predicted bounding box coordinates), object loss (ensuring accurate object presence prediction), and classification loss (optimizing object categorization accuracy). These loss functions work together to guide the training process of our object detection model, striving for improved accuracy in object localization, object classification, and the distinction between objects and background regions.

3.5.1 Box Loss

The box loss, as defined in Equation 5 is the difference between the predicted bounding box parameters (like center coordinates, width, and height) and the actual ground truth parameters of the boxes. The objective of minimizing box loss is to improve the precision of the model in localizing objects within an image.

$$\text{Box Loss} = \sum_{i=0}^{S^2} \sum_{j=0}^B \text{obj}_{ij}^{(k)} \cdot [(x_{ij} - \hat{x}_{ij})^2 + (y_{ij} - \hat{y}_{ij})^2 + (\sqrt{w_{ij}} - \sqrt{\hat{w}_{ij}})^2 + (\sqrt{h_{ij}} - \sqrt{\hat{h}_{ij}})^2] \quad (5)$$

Here, (x_{ij}, y_{ij}) and $(\hat{x}_{ij}, \hat{y}_{ij})$ are the predicted and true bounding box center coordinates, while $w_{ij}, h_{ij})$ and $(\hat{w}_{ij}, \hat{h}_{ij})$ are the predicted and true bounding box width and height respectively. S^2 represents the number of grid cells, in $S \times S$ grid and B is the number of bounding boxes predicted per grid cell, $\text{obj}_{ij}^{(k)}$ is the boolean indicator of whether object k exists in grid cell (i, j) .

Depicted by the blue curve in Figure 8, the training box loss offers insights into the box loss trajectory during the model’s learning phase. An evident reduction in both training box loss and validation box loss as epochs progress denotes the model’s sharpening ability in bounding box predictions throughout its training epoch cycles.

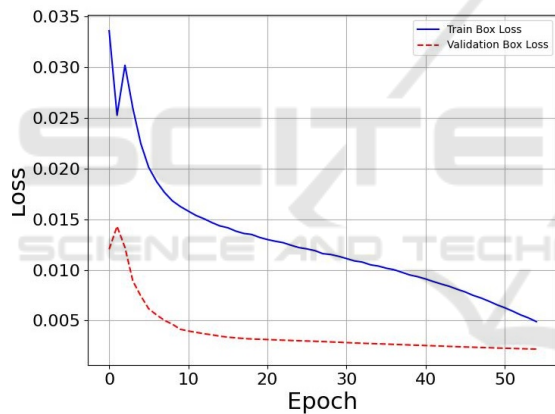


Figure 8: Box Loss Curves.

3.5.2 Object Loss

Object Loss as shown in Equation 6 is computed using the mean squared error (MSE) between the predicted confidence scores and the ground truth confidence scores where the confidence scores confidence score indicates whether an object is present in a given grid cell and how accurate the bounding box is.

$$\text{Obj Loss} = \sum_{i=0}^{S^2} \sum_{j=0}^B \text{obj}_{ij}^{(k)} \cdot (\text{Conf}_{ij} - \hat{\text{Conf}}_{ij})^2 \quad (6)$$

Here, Conf_{ij} is the predicted confidence score, and $\hat{\text{Conf}}_{ij}$ is the true confidence score.

Figure 9 illustrates a concurrent decline in both training and validation object loss as training progresses, indicating an enhancement in the model’s de-

tection capabilities. The absence of a significant deviation between these two metrics throughout the training epochs reassuringly suggests that the model is not afflicted by issues such as over-fitting.

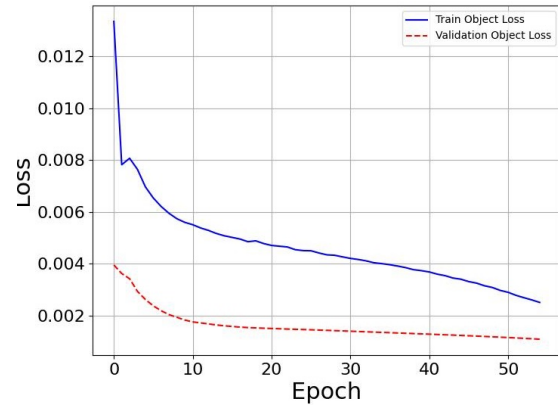


Figure 9: Object Loss Curves.

3.5.3 Classification Loss

The classification loss is defined as the misalignment between the predicted class probabilities and the actual binary class labels within the training dataset, measuring the model’s reliability in associating objects with their true categories. As visualized in Figure 10, an analogous declining trend observed in both training and validation classification loss across the training epochs not only corroborates the model’s improving proficiency in object classification but also mirrors the previously discussed trends in box and object loss, solidifying the consistency in the model’s learning and adaptation throughout the training process.

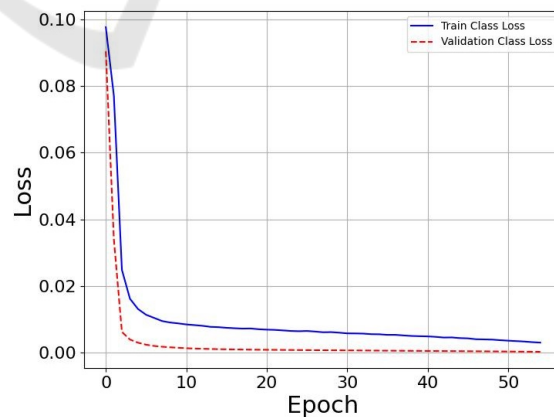


Figure 10: Classification Loss Curves.

4 CONCLUSION

In conclusion, this paper introduced a well-rounded approach toward automated military badge detection on government documents by utilizing the YOLOv5 model. By innovatively automating the training data labelling process and generating a simulated dataset of military and official documents, circumventing the issue of public unavailability, a scalable and precise badge detection system was established. Through strategic training and hyper-parameter tuning, the YOLOv5 model showcased substantial proficiency in detecting various badge types within the documents, illustrating a promising stride in document-based object detection.

REFERENCES

- Arvind, N. (2023). A semi-automatic method for document classification in the shipping industry. In *Proceedings of Neptune's conference, Samudramanathan 2023 IIT Kharagpur*.
- Bailey, E. S., Bonnici, A., and Cristina, S. (2022). A cascaded approach for page-object detection in scientific papers. In *Proceedings of the 22nd ACM Symposium on Document Engineering, DocEng '22*, New York, NY, USA. Association for Computing Machinery.
- Brown, J. D. (2010). Developing an automatic document classification system: A review of current literature and future directions. Technical Memorandum DRDC Ottawa TM 2009-269, Defence Research and Development Canada.
- Chiu, P., Chen, F., and Denoue, L. (2010). *Picture detection in document page images*.
- Ciecierski, K. and Kamola, M. (2020). *Comparison of Text Classification Methods for Government Documents*, pages 39–49.
- Deng, Q., Ibrayim, M., Hamdulla, A., and Zhang, C. (2023). The yolo model that still excels in document layout analysis. Preprint under review at Signal, Image and Video Processing as of August 2023.
- Forczmański, P., Smolinski, A., Nowosielski, A., and Małeckki, K. (2020). *Segmentation of Scanned Documents Using Deep-Learning Approach*, pages 141–152.
- Henderson, P. and Ferrari, V. (2017). End-to-end training of object class detectors for mean average precision. In Lai, S.-H., Lepetit, V., Nishino, K., and Sato, Y., editors, *Computer Vision – ACCV 2016*, pages 198–213, Cham. Springer International Publishing.
- Huber-fliflet, N., Wei, F., Zhao, H., Qin, H., Ye, S., and Tsang, A. (2019). Image analytics for legal document review : A transfer learning approach. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4325–4328.
- Kallemudi, G., Hashmi, K. A., Pagani, A., Liwicki, M., Stricker, D., and Afzal, M. Z. (2022). Toward semi-supervised graphical object detection in document images. *Future Internet*, 14(6).
- Li, M. and Ghosal, S. (2014). Bayesian Multiscale Smoothing of Gaussian Noised Images. *Bayesian Analysis*, 9(3):733 – 758.
- Liu, C., Tao, Y., Liang, J., Li, K., and Chen, Y. (2018). Object detection based on yolo network. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 799–803.
- Orosz, T., Vági, R., Csányi, G. M., Nagy, D., Üveges, I., Vadász, J. P., and Megyeri, A. (2022). Evaluating human versus machine learning performance in a legal-tech problem. *Applied Sciences*, 12(1).
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.
- Rezkiani, K., Nurtanio, I., and Syafaruddin (2022). Logo detection using you only look once (yolo) method. In *2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, pages 29–33.
- Sarwo, Heryadi, Y., Abdulrachman, E., and Budiharto, W. (2019). Logo detection and brand recognition with one-stage logo detection framework and simplified resnet50 backbone. In *2019 International Congress on Applied Information Technology (AIT)*, pages 1–6.
- Song, Y., Li, Z., He, J., Li, Z., Fang, X., and Chen, D. (2019). Employing auto-annotated data for government document classification. In *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence, ICIAI '19*, pages 121–125, New York, NY, USA. Association for Computing Machinery.
- Suto, J. (2023). Improving the generalization capability of yolov5 on remote sensed insect trap images with data augmentation. *Multimedia Tools and Applications*.
- Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.