https://doi.org/10.2352/CIC.2023.31.1.12

Metrology-Driven Image Synthesis for Quality Control

Meldrick Reimmer, Hermine Chatoux, Olivier Aubreton; ImViA, Université de Bourgone; Dijon, France

Abstract

Metrology plays a critical role in the rapid progress of Artificial Intelligence (AI), particularly in computer vision. This article explores the importance of metrology in image synthesis for computer vision tasks, with a particular focus on object detection for quality control. The aim is to improve the accuracy, reliability and quality of AI models. Through the use of precise measurements, standards and calibration techniques, a carefully constructed dataset has been generated and used to train AI models. By incorporating metrology into AI models, we aim at improving their overall performance and robustness.

Introduction

The rapid advancement of artificial intelligence (AI) has had a revolutionary impact on various domains especially computer vision [1, 2, 3, 4]. Image synthesis is a key component of computer vision. It plays a central role in applications such as computer graphics, virtual reality, and object recognition.

Ensuring the reliability, accuracy and quality of synthesised images is a challenging task in computer vision. The integration of metrology, the science of measurement, has emerged as a valuable approach to overcome these challenges. Study in [5], highlight the role of metrology in improving the accuracy and reliability of computer vision systems through precise calibration techniques. A study in [6], highlight its importance in achieving high perceptual quality by emphasising the need for accurate measurements and adherence to standards. Generative Adversarial Networks (GANs) [7] have also emerged as the most widely used method for image synthesis [8, 9, 10]. However, while GANs excel at capturing visual patterns and generating visually appealing images, they often struggle to incorporate the underlying physics or intrinsic properties of the target objects. This limitation is the reason for the omission of GAN in this study.

Building on the findings in [5, 6, 11], this study aims to explore the importance of metrology in improving the accuracy, reliability, and quality of AI models for image synthesis in computer vision tasks for quality control depicted in Figure 1. The integration of precise measurements, adherence to standards, and calibration techniques is crucial to produce images that closely match the physical properties and characteristics of real-world objects.

In this research study, we present a comprehensive pipeline that integrates both radiometric and geometric metrics into the image synthesis process. Three types of datasets were created to train three AI models. This research aims to provide compelling evidence of the critical role of accurate measurements in optimising the effectiveness of AI models.

In the following sections, the methodology used in this study will be discussed, outlining the procedures for acquiring the necessary measurements. This is followed by a discussion of image synthesis and dataset generation in the next section. In the following section, we will discuss the training of an AI model, followed by a comprehensive overview of the experimental protocol. We will then present and analyse the results obtained, providing substantial evidence in support of our theory. Finally, we will conclude.

Metrology of the scene

This section presents the methodology used in this study, which aims to demonstrate how more effective an AI model becomes in computer vision tasks, specifically object detection, when the metrology of the targeted object are known. This approach starts with a comprehensive analysis of the geometric and radiometric metrics of the camera used to capture the objects in the scene. In addition, we measured the reflectance and fixed the shape of the objects.

Geometric and Radiometric Calibration

A Nikon D-850 camera was used in this study. The calibration process was based on the methodology introduced in [11]. It involved two main steps: characterising the sensor's spectral sensitivity function (SSF), and performing a geometric calibration.

Camera Characterisation

In this study, Zhang's [12] geometric calibration method was used, in which multiple images were taken of a known calibration pattern, such as a chessboard. The aim was to establish a precise mapping between the 2D image coordinates (x_i, y_i) and the corresponding known 3D object coordinates (X_i, Y_i, Z_i). By placing the calibration pattern in different positions and orientations, we ensured comprehensive coverage of the camera's field of view, allowing for a robust calibration process.

To determine the intrinsic parameters, we focused on the camera matrix, which relates the homogeneous image coordinates (u, v, w) to the 3D object coordinates (X, Y, Z):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

Where, s denotes the scaling factor, (f_x, f_y) represent the focal lengths, and (c_x, c_y) indicate the principal point coordinates.

To estimate the internal camera parameters, we solved a set of equations based on the correspondence between image and object coordinates. The refinement of the camera matrix parameters involved minimising the reprojection error. This error quantifies the disparity between the projected image points and the actual image points.

The precise calibration of the camera ensured a highly accurate measurement of both geometric and radiometric properties throughout the experiments.

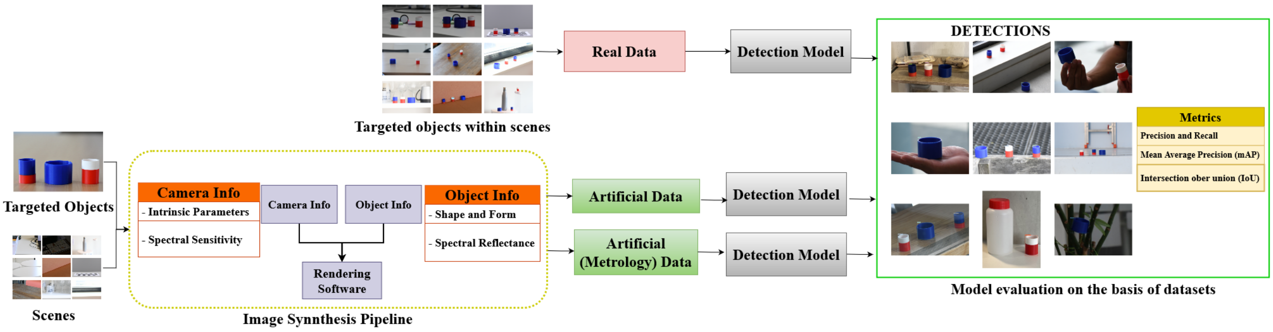

Figure 1. *An overview of the proposed pipeline, in which images in three forms (real data, artificial data and artificial with metrology data) are collected and used to train detection models, and the models are then evaluated against given metrics to see which perform best with a given set of data.*

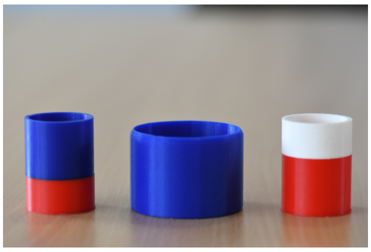

Figure 2. *An image of the 3D-printed objects*

To measure the SSF of the camera, we used a monochromator. This allows us to measure the sensitivity of each channel per wavelength. The experimental setup covered a wavelength range of 360-750 nm with a spectral resolution of 10 nm. The image integration time was 5 s.

The measured SSF at a given wavelength is:

$$C_k(\lambda) = \frac{\text{mean value of pixels}}{\text{exposure time}}, \tag{2}$$

where C_k is the SSF of channel k and λ is the wavelength.

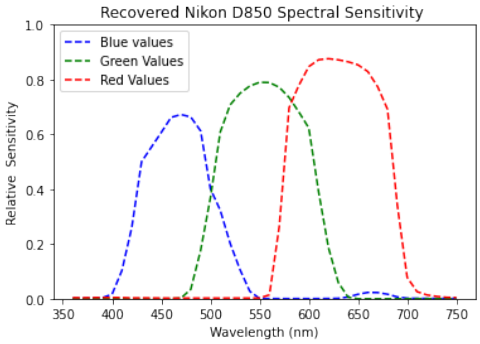

Figure 3. *Spectral Sensitivity of Nikon D-850: Wavelength Range 360-750 nm, Spectral Resolution: 10 nm, Image Integration Time: 5s.*

Object Characteristics

Three cylindrical shaped objects were modeled and 3D printed with three different colours (blue, red and white) as seen in Figure 2. The characteristics of these objects, were also measured. Specifically, the spectral reflectance of the objects using a CS-1000 spectrophotometer in a D65 light cabinet, following a procedure similar to that described in [11]. This provided valuable data on the reflectance properties of the objects Figure 4, in knowing how light interacts with the object's surface to have a lustrous appearance of objects in a virtual scene.

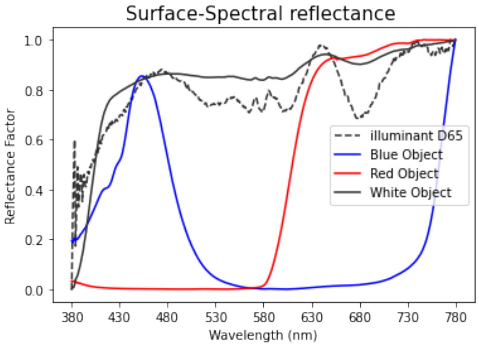

Figure 4. *Measured spectral reflectance of the targeted objects and illuminant radiance.*

Experimental Protocol

Armed with this meticulous knowledge, we proceed to generate synthetic images of the objects under study. These synthetic images are then used as training data to meticulously train a Convolutional Neural Network (CNN) specifically tailored for object detection.

Dataset Generation

Three dataset consisting of three different types of data were constructed. The first two types were generated using the rendering software Blender [13], one including the relevant obtained metrics and the other excluding them. The third type of data was acquired by capturing the target objects in different scenes using the measured RGB sensor, increasing the diversity and robustness of the dataset.

This collected data is used to train three separate models. By using the synthetic images generated by Blender and the real-world images captured, we aim to develop models that can effectively analyse and detect the objects in the scenes.

The blender tool facilitated the incorporation of custom parameters through a third-party Python API. Leveraging this integration, we were able to use the Spectral Sensitivity Function (SSF), derived from the Eq.(2) and shown in Figure 3. This use of Blender's capabilities allowed us to achieve precise control and flexibility in the image synthesis process, ensuring accurate representation of the desired properties in the synthesised images.

In generating the images, the first step done was to model the real camera into the virtual scene, so the camera matrix measured was converted from pixels to millimeters (mm) to be able to use in blender, after which we performed the following process, illustrated in Figure 5 :

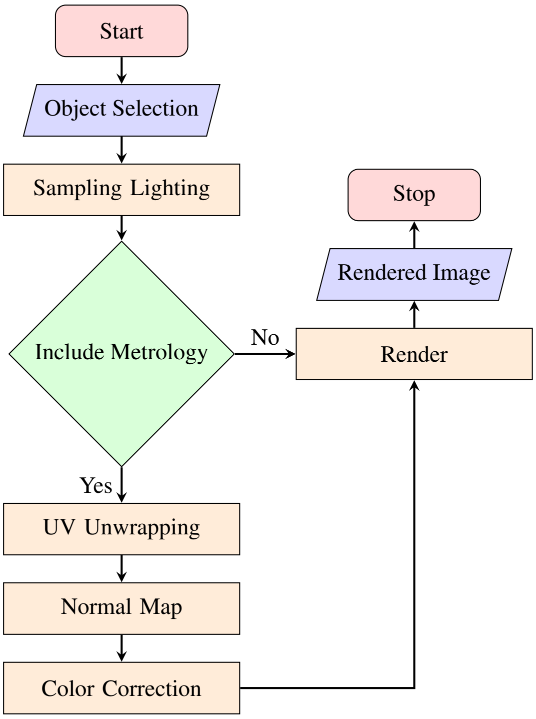

Figure 5. *A Flowchart depicting the precise workflow for artificially generating visually realistic images with or without geometric and radiometric metrics. Decision point allows for integration of acquired metrics.*

- **Object Selection:** each object is selected for precompositing, and a new material is created for the object in the Material Properties tab, giving it a unique name.
- **Sampling Lighting:** a blackbody node is added to the material to sample the lighting in the scene and adjust the object's colors accordingly, simulating how light interacts with the object's surface.
- **UV Unwrapping:** after obtaining the texture map for the objects, an image-based texture UV mapping [14, 15] is performed to project a 2D texture of a given image onto the surface of the object in 3D.
- **Normal Map:** a normal map is added to enhance the realism of the object, including surface details such as bumps and ridges.
- **Color Correction:** a color ramp node is added to the object's material, allowing for color adjustment using the derived spectral sensitivity function values.
- **Render:** the Blender Cycles (Ray tracing) engine is used to render the virtual scenes in order to realistically simulate the lighting of a scene and its objects.

A total of 60 images with the targeted object were each generated. Samples of image acquisitions and generated can be seen in Figure 6.

Dataset Evaluation

Objective metrics were used to quantitatively measure image similarity and quality of the generated images we took into account both low-level and high-level visual features. Since there was a referenced image to compare, metrics used in the evaluation were:

1. **Structure Similarity Index Measure (SSIM) [16]**:

   SSIM compares the similarity between the original image and the generated image, considering lighting, contrast, and structure. It is calculated using the following formula:

   $$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \qquad (3)$$

   where x and y are the original and generated images. μ_x and μ_y are the means of x and y. σ_x^2 and σ_y^2 are the variances of x and y. σ_x^2 and σ_y^2 are the variances of x and y. σ_{xy} is the covariance of x and y. C_1 and C_2 are constants to avoid division by zero. SSIM provides a value between -1 and 1, where 1 indicates that the images are identical.

2. **Peak Signal to Noise Ratio (PSNR) [17]**:

   PSNR measures the ratio between the maximum possible power of an image and the power of the noise present. It is calculated using the following formula:

   $$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \qquad (4)$$

   where MAX is the maximum possible pixel value (e.g., 255 for an 8-bit image).MSE is the mean squared error between the original and generated images. PSNR is expressed in decibels (dB), and a higher PSNR value indicates better image quality.

3. **Average** (ΔE_{2000}):

   The average color difference (ΔE_{2000}) was used to assess the color difference between the original and generated images. It was calculated using the CIEDE2000 formula [18], which involves complex calculations based on the Lab color space.

Table 1, provides an evaluation of the quality and fidelity of the rendered objects. This assessment was conducted using a single reference image. The objects are categorized into two

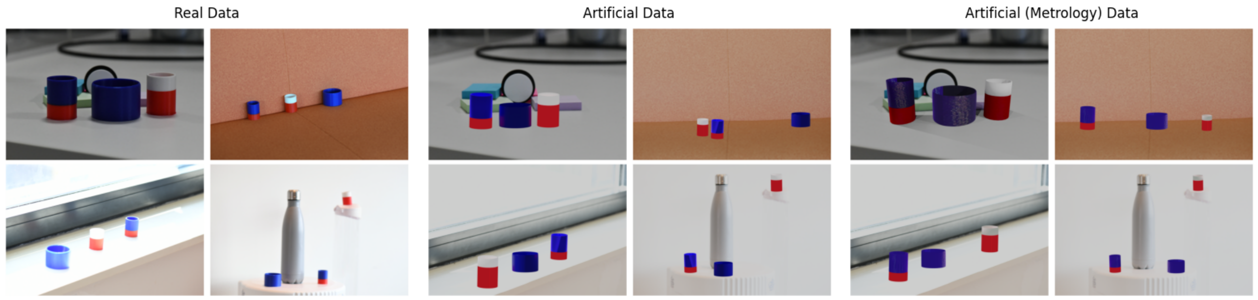

Figure 6. *Illustrating samples of image acquisitions with Nikon D-850real data, artificial data and artificial (metrology) data collected and used in this study.*

Table 1: Assessing the quality and fidelity of rendered objects.

Objects		SSIM	PSNR	Avg. ΔE_{2000}
Red & White	Artificial	**0.87**	**34.7**	13.4
	Metrology	0.84	32.4	**10.4**
Blue	Artificial	0.67	31.8	19.3
	Metrology	**0.7**	**33.6**	**11.5**
Blue & Red	Artificial	**0.65**	31.6	19.3
	Metrology	0.63	**39**	**10.8**

groups: artificial and metrology, representing different synthesis approaches.

The average (ΔE_{2000}) shows minimal values throughout, highlighting a better colour fidelity achieved between the synthetically generated metric objects and their real-world counterparts. This fidelity is further supported by the achievement of PSNR values of over 30dB and an average SSIM of over 60, indicating a high level of image quality. However, it is important to note the conventional interpretation of ΔE_{2000} in different areas, where ΔE_{2000} values greater than 10 typically indicate a significant colour difference. In the context of this study, ΔE_{2000} values greater than 10 do indeed indicate noticeable colour differences between the reference and generated images.

A reduction in SSIM can be observed for the Red & White and Blue metrics, accompanied by higher values for Colour Difference. This can be attributed to the influence of the UV unwrapping image-based texture approach, which introduces a natural noise component that affects the SSIM. However, this approach gives commendable results in terms of PSNR, colour difference, the synthesised targets with metrology have distinct and very similar characteristics to their real-life counterparts.

Model Selection

The selection of an optimal object detection model for this study was carefully considered given the unique characteristics of the collected data, particularly its metrological nature. With these requirements, YOLOv7 [19] was chosen as the ideal candidate due to its remarkable ability to exploit intricate details and specific features within the dataset, while delivering state-of-the-art accuracy and speed.

We fine-tuned the YOLOv7 model using pre-trained weights from the COCO dataset [20], a comprehensive collection of labelled diverse object images. YOLOv7 architectures were used, which consists of 37 million parameters and achieves more than 51% mean average precision (mAP). The 3 sets of datasets, which consists of 3 classes were utilized to train these architectures, resulting in the development of 3 models seen in Table 2. Each model was fine-tuned to effectively exploit the distinct characteristics of its corresponding dataset.

The hyperparameter optimization techniques applied to enhance the performance and effectiveness of these models are Batch Size: 2; Epoch: 100; Optimizer : Adam; Learning rate: 0.01; Size: 1280 × 1280.

Data Collection and Model Architecture

The model training procedure involved a complex dataset of exactly 60 images. This was carefully divided into two subsets: a training set of 42 images and a validation set of 18 images. In addition, a special collection of 30 never-before-seen images was carefully reserved for rigorous testing of the trained models.

Significantly, the images within the training set depicted the target objects in a variety of contextual scenarios, as illustrated in Figure 6. In contrast, the unfamiliar test dataset introduced a new dimension by including scenarios in which target objects were partially obscured by various occluding elements, including human fingers and various objects. This deliberate inclusion of occluded scenarios in the test dataset served the purpose of a careful evaluation, assessing the resilience and adaptability of the models in the face of challenging real-world conditions.

Table 2: Models based on YOLOv7 architectures

Model	Architecture	Data
1	YOLOv7	Real
2	YOLOv7	Artificial
3	YOLOv7	Artificial (Metrology)

Results and Discussions

To evaluate the performance of the three trained models, a rigorous benchmarking process was conducted using the test image dataset. A meticulously annotated ground truth dataset was prepared, comprising 30 images with a total of 65 annotations. These annotations were categorized into three classes: ”Red and White” (21 annotations), ”Blue and Red” (18 annotations), and ”Blue” (26 annotations).

Various evaluation metrics were employed to assess the models, including precision, recall, mean average precision (mAP) at IoU (Intersection of Union) thresholds ranging from 0.3 to 0.5, and the F1 score. These metrics provided comprehensive insights into the models' performance in object detection and classification tasks.

Table 3: Models performance

Model	Precision	Recall	F1 Score	mAP
1	58%	**72%**	0.65	69%
2	37%	5%	0.44	41%
3	**91%**	66%	**0.76**	**79%**

Table 3, presents a comparison of the performance metrics of the trained models used in this study. model 1 has 58% correct detection and successfully identifies 72% of the relevant objects. The calculated F1 score of 0.65 reflects the overall performance of the model. In addition, the model achieves an mAP of 69%, indicating its consistent performance across different object classes.

On the other hand, model 2 has a lower precision of 0.376, indicating a higher rate of false positives. 50% of the relevant objects are missed by the model 2. The F1 score of 0.44, highlighting the suboptimal performance of the model. The mAP value of 41% indicates a poor localisation and identification of objects.

Finally, model 3 shows excellent performance with a precision of 0.91, effectively minimising false positives. The recall of 0.66 indicates successful identification of a significant proportion of relevant objects. The calculated F1 score of 0.76 highlights the model's balanced trade-off between precision and recall. Notably, model 3 achieves the highest mAP value of 79%, confirming its effectiveness in accurately identifying target objects.

To further explore the evaluation of model performance, we performed a careful analysis of the confusion matrices associated with model 1 and model 3. These matrices, shown in Figure 8 and Figure 9 respectively, provide a visual representation of true positives, true negatives, false positives and false negatives. They provide valuable insight into the capabilities and limitations of these models.

In the context of white and red objects, model 1 showed an impressive true positive rate of 85%. Remarkably, it showed minimal false positives, with the exception of a 15% incidence of background false negatives. In stark contrast, model 3 showed a true detection rate of 62%, accompanied by a notable 37% incidence of background false negatives. This result is consistent with our findings in Table 1.

Turning our attention to blue and blue-red objects, we observed that true positives were more common in the metrics data than in the real images. This difference highlights the superior accuracy and reliability of model 3, underlining its ability to accurately identify target objects.

However, it is worth noting that a significant number of blue objects eluded detection by most models. This observation suggests that the complexity inherent in the proposed images posed a formidable challenge for the models to effectively detect the blue objects. For visual confirmation, see Figure 7. These images were deliberately designed to rigorously test the robustness of the models.

In the light of this extensive analysis, it is clear that model 3 achieved commendable mean average precisions (mAPs). This success can be attributed to its ability to detect objects across different classes, coupled with minimal false predictions and impressive true positive rates.

However, a closer examination of the confusion matrices reveals a common challenge for both model 1 and model 3, namely accurately distinguishing target objects from the background. In the case of model 1, we identified instances where background false negatives occurred, indicating an occasional failure to detect target objects against the background.

For model 3, this problem of background false negatives is more pronounced, indicating a higher frequency of missed detections against the background screen. This observation highlights an area for improvement in both models, particularly in scenarios where accurate discrimination between target objects and background elements is paramount.

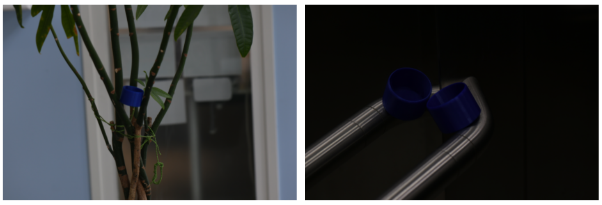

Figure 7. Showing an example of complex scenes with blue object.

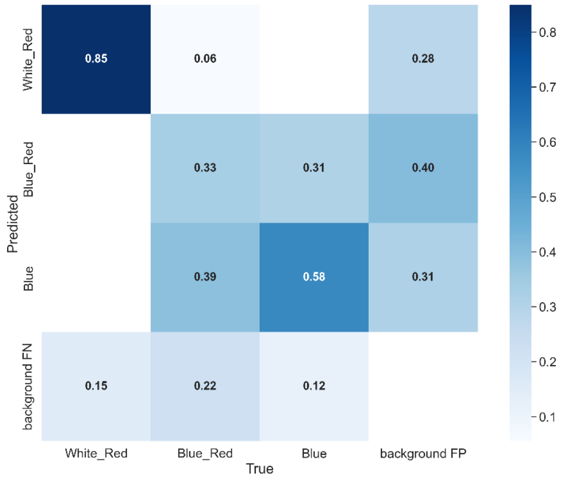

Figure 8. Illustrating the confusion matrix of model 1.

A further analysed of the performance of the models at two different confidence thresholds, 3 and 5, as shown in Table 4.At confidence level 3, both model 1 and model 3 showed superior capabilities compared to model 2. In particular, they showed higher precision in detecting a greater number of target objects in all three different target classes. However, increasing the threshold to 5, which represents a more stringent confidence criterion for positive detections, revealed a challenge for the models in accurately identifying target object classes. A notable exception

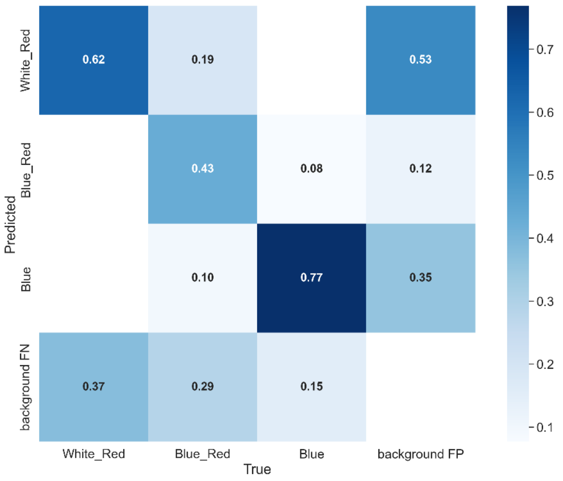

Figure 9. *Illustrating the confusion matrix of model 3.*

Table 4: Model Performance Comparison @ threshold

Threshold	Class	Ground Truth	Model		
			1	2	3
3	Blue	26	**24**	9	21
	Red_white	21	**16**	0	12
	Blue_Red	18	**14**	1	13
5	Blue	26	7	1	**15**
	Red_white	21	**9**	0	**9**
	Blue_Red	18	0	0	**4**

was model 3, which consistently delivered exceptional results and demonstrated robust detection capabilities, maintaining superior accuracy even at this more demanding threshold, reflecting an increased level of confidence in its detection results.

It's worth noting that model 2, trained on synthetically generated data, showed comparatively suboptimal performance. This performance gap can be attributed to the inherent nature of the synthesis process. Unlike real images, metrological synthesis produces noise-free images, allowing the model to focus exclusively on characterising relevant information.

These insightful findings underscore the remarkable effectiveness of model 3 in consistently demonstrating high levels of confidence in object recognition tasks. Model 3, built on a comprehensive YOLOv7 architecture and trained with carefully generated measurement data, demonstrated an exceptional ability to recognise and accurately characterise the distinctive features of the target objects.

Conclusion

This study demonstrates the potential of using synthetic images based on metrology to improve AI models for object recognition tasks. The results highlight the strong generalisation capabilities of the model when trained on synthetic data with metrology. The near perspective is to work on the uv unwrapping based on image texture, to improve the detection of bi-colour objects.

Further improvements in attention mechanisms and simplification of narratives may lead to even better results. In industrial quality control scenarios, where attention is primarily focused on specific objects, the proposed method may perform better. The idea of co-learning, combining synthetic and real data, presents a promising approach to continually enhance AI models in object recognition tasks. By leveraging the strengths of both synthetic and real-world data, researchers and practitioners can ensure that AI models remain adaptable and accurate in different contexts.

References

[1] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37).

[5] Golnabi, H., Saadatseresht, M., & Madani, M. (2018). A review of radiometric and geometric calibration methods in computer vision. Journal of Applied Remote Sensing, 12(2), 025004.

[6] Wang, X., Zhang, L., & Bovik, A. C. (2020). Fidelity perception and metrology: Bridging the gap. IEEE Signal Processing Magazine, 37(1), 84-98.

[7] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems. 2014. p. 2672–80.

[8] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. CoRR, abs/1511.06434.

[9] Karras, T., Laine, S., & Aila, T. (2018). A Style-Based Generator Architecture for Generative Adversarial Networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4396-4405.

[10] Brock, A., Donahue, J., & Simonyan, K. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. ArXiv, abs/1809.11096.

[11] Reimmer Meldrick, Chatoux Hermine, Olivier Aubreton. Improving the Fidelity of Synthesized Images by Integrating Real-World Data. Traitement et l'Analyse de l'Information : Méthodes et Applications, TAIMA-2023, Arts-pi, May 2023, Hammamet, Tunisia. pp.236 - 249. (hal-04189868).

[12] Zhang, Zhengyou. "A Flexible New Technique for Camera Calibration." IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000): 1330-1334.

[13] Blender Online Community.Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[14] Pelechano, N., & Gutierrez, D. (2006). Automatic generation of textures for real-time 3D meshes. In Proceedings of the 13th Eurographics Symposium on Rendering (pp. 213-224).

[15] Zhang, S., & Bala, K. (2010). Automatic texture atlas generation using modified greedy heuristic. In Proceedings of the ACM SIG-

GRAPH Symposium on Interactive 3D Graphics and Games (pp. 77-84).

[16] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600-612.

[17] A. Horé and D. Ziou, ”Image Quality Metrics: PSNR vs. SSIM,” 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 2366-2369, doi: 10.1109/ICPR.2010.579.

[18] Sharma, G., Wu, W., & Dalal, E. N. (2005). The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. Color Research & Application, 30(1), 21-30.

[19] Wang, Chien-Yao & Bochkovskiy, Alexey & Liao, Hong-yuan. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 10.48550/arXiv.2207.02696.

[20] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In European conference on computer vision (pp. 740-755). Springer.